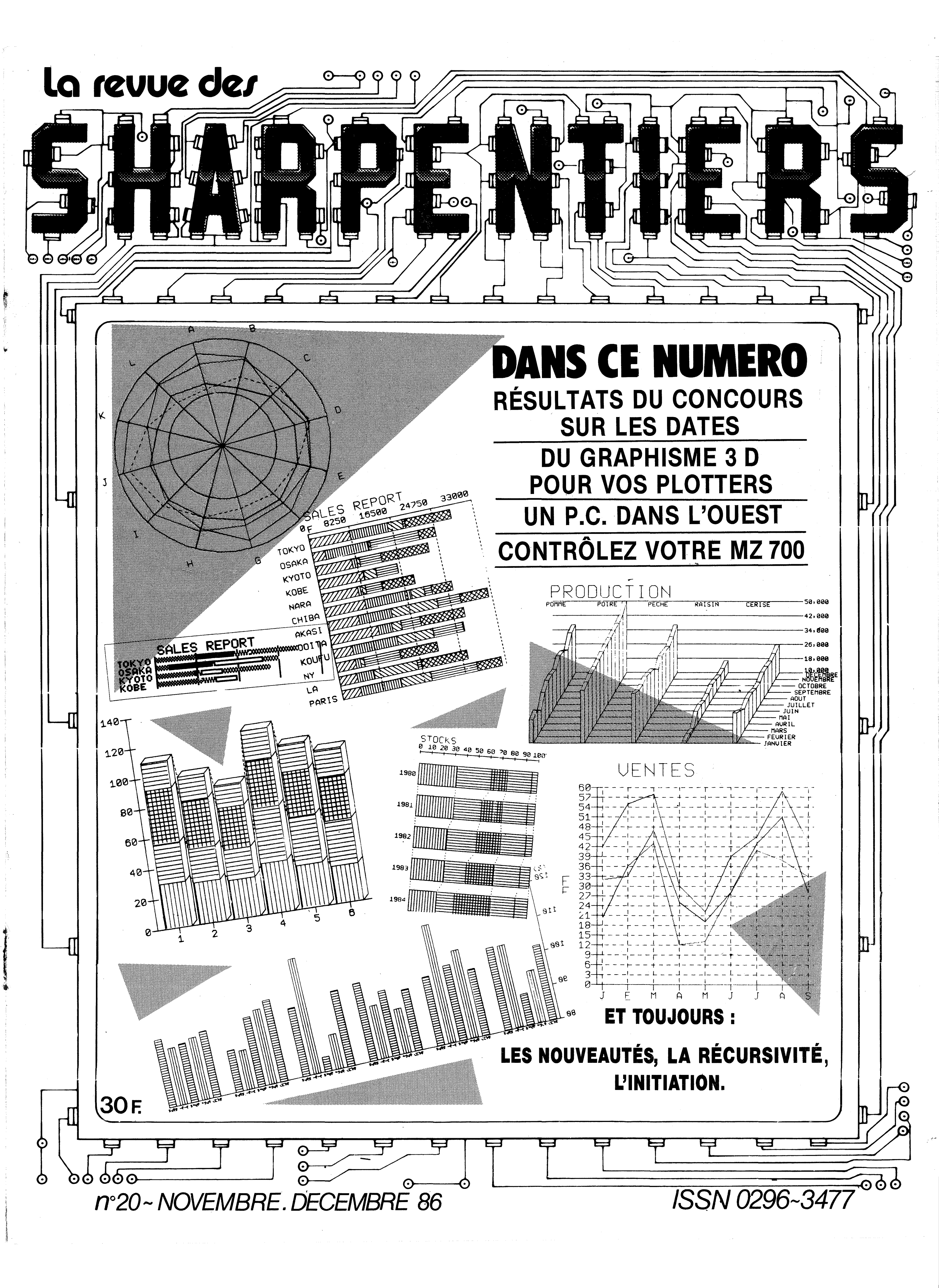

La revue des
SHARPENTIERS
DANS CE NUMERO
RÉSULTATS DU CONCOURS SUR LES DATES
DU GRAPHISME 3 D POUR VOS PLOTTERS
UN P.C. DANS L'OUEST
CONTRÔLEZ VOTRE MZ 700
ET TOUJOURS :
LES NOUVEAUTÉS, LA RÉCURSIVITÉ, L'INITIATION.
SALES REPORT
PRODUCTION
STOCKS
VENTES
30 F.
n°20 ~ NOVEMBRE . DECEMBRE 86
ISSN 0296~3477

LE SHARPENTIER
N°20

RÉDACTEUR EN CHEF
Marc GIRONDOT

RÉDACTEURS
Luc BURELLER
Simon CHAGNOUX
Christophe POULIN
Jean-François VIGNAUD

SECRÉTAIRE DE RÉDACTION
Dominique DUBAN

ONT COLLABORÉ A CE NUMÉRO
Jacques ANDRÉ
Éric ARISTIDI
Frédéric BLONDIAU
R. BOURION
Albert BURGERT
Bruno CAPLA
M. DALLET
Klaus DITZE
M. DOIDY
Norbert ENNULAT
Didier GUTIERREZ
Michel HOUDARD
Bernard KOKANOSKI
J.-C. MOREL
Nicolas NOUAIL
Alain REBSAM
Olivier SCHANEN
M. TOUSSAINT
Patrice VOISIN

RÉALISATION
IN QUARTO
19, rue Frédéric-Lemaître
75020 PARIS

PUBLICITÉ
Jean-François VIGNAUD

La revue des Sharpentiers est éditée par le Club des Sharpentiers.

Commission paritaire en cours

Dépôt légal à parution

Ce numéro a été tiré à 3 000 ex.

EDITO

Le numéro 20 : à nouveau, un bulletin dans les temps. Les nombreux problèmes que nous avons connus ces derniers temps ont fait que le numéro 19 a eu pratiquement deux mois de retard. Il nous a fallu changer d'outil pour gérer le fichier adhérents (l'ancien nous ayant laissé tomber), trouver le temps de trier, mettre sous enveloppe et faire expédier votre bulletin, et ce dans des conditions très précaires. Désormais, cette gestion sera facilitée car le club est à nouveau regroupé en un seul lieu. Depuis le 1er novembre, l'adresse du club est la suivante :

CLUB DES SHARPENTIERS
***c/o* MICRO-ARCHI**
79, rue du Temple, 75003 PARIS
(joignez une enveloppe timbrée pour la réponse)

Notre numéro de téléphone est le suivant : (16) 1 42.74.07.68.
Des permanences seront assurées en semaine (téléphonez au numéro ci-dessus pour connaître les dates et horaires d'ouverture du local).

Le club, c'est de nouveaux services à la mesure de nos moyens et des aides que l'on reçoit.
C'est plus de permanences, mais c'est aussi un serveur vidéotex dans lequel vous trouverez de nombreux renseignements sur vos machines et sur le club (accédez souvent à ce service, que nous utiliserons pour vous faire connaître rapidement des actions ponctuelles).
Vous y trouverez aussi une messagerie qui vous permettra de nous joindre rapidement. Les modalités d'accès à ce service vous sont données dans les pages suivantes. Des explications complètes feront l'objet d'un article dans le bulletin n° 21.
Parlez du club autour de vous, après une courte période de problèmes et de stagnation, le club renaît. Si vous le pouvez, venez nous aider à faire vivre le club. Venez nombreux à nos réunions.
A dans deux mois, nous nous retrouverons dans le bulletin n° 21 le 15 janvier 1987.

Luc BURELLER

LES NOUVEAUTÉS

Comme tous les ans, le SICOB d'automne nous amène sa moisson de nouveaux produits. La gamme des PC SHARP se retrouve ainsi presque complètement rénovée. La plupart des modèles existant à ce jour se voit remplacé par une machine équivalente avec un nouveau «look» ou des performances supérieures pour un coût identique. Les extensions disponibles (lecteur de disquette, carte mémoire) expliquent ce changement. Normalement, à l'heure où vous lirez ces lignes, ces PC devraient être commercialisés.

PC 1246 S	: c'est un PC 1246 avec une nouvelle présentation
PC 1248	: 1246 S avec 8 Ko (enfin) !
PC 1262	: identique au 1261 (apparemment).
PC 1403	: à mi-chemin entre le 1401 et le 1402 : écran de 24 caractères, 8 Ko, calcul matriciel, interface disquette (CE-12FD).
PC 1425	: modèle scientifique surtout orienté Statistiques : carte Mev, 1 ligne de 24 caractères, lecteur de disquette (CE-12FD) en option.
CE-2H32M	: carte mémoire petit format de 32 Ko.
CE-1601N	: lecteur optique de codes à barres pour PC 1600. Se branche à la place de la fibre optique.
CE-12CAR	: alimentation voiture (allume-cigare) pour PC 1600.



TOUJOURS SOFT TEXT...

Il m'arrivait d'avoir plusieurs textes à écrire et il n'est pas prévu de les sauvegarder ou de les imprimer en plusieurs parties. L'auteur ayant gentiment laissé de la place à la fin du BASIC, les modifications à apporter se réaliseront facilement. Modifier le programme BASIC selon les données suivantes :

```
600 :"L"CURSOR : PRINT "1=LEC
     2=FUS  3=SAV  4=SAP":
     A=VAL INKEY$ : IF A <1 OR A>
     4 THEN "L"
607 :IF A=4 GOSUB 950 : D=F :
     GOSUB 950 : L=F— D : GOTO
     650
950 :CALL 256*AD+&C6 : BEEP 1 :
     F=PEEK  &78E4*256+PEEK
     &78E5 : RETURN
```

UTILISATION DE LA NOUVELLE FONCTION

Le texte étant entré, choisir sur le menu CAS puis SAP (sauvegarde partielle). Entrer le nom du fichier et placer le magnétophone sur enregistrement. Après un ENTER, vous vous retrouvez sous l'éditeur. Placer alors le pavé clignotant sur le début du texte à sauver. Appuyer sur OFF comme pour sortir de l'éditeur. Après le BEEP, placer le curseur sur la fin du texte à sauver et appuyer sur la touche OFF.

Le texte sera sauvegardé partiellement sur votre K7.

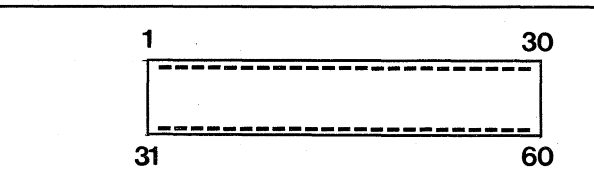
A l'aide de cette possibilité de sauvegarde partielle, vous pourrez aussi intervertir des paragraphes puisque la fonction FUS est conservée.

M. GUTIERREZ Didier



POUR LES FANATIQUES... DU FER A SOUDER

Voici le schéma exact de la prise QCN 60 broches du PC 1500 car elle n'est pas décrite dans le TRM. Pour savoir quels sont les signaux aux broches, voyez votre TRM.



Suite à la réclamation d'un adhérent, nous conseillons à toute personne intéressée par le connecteur QCN de se procurer le TRM pour savoir au moins quels sont les signaux aux broches du connecteur. D'ailleurs à ce propos, sur la liste dù connecteur 60 broches page 103, il faut inverser DME1 et DMEO. DME1 est donc en 56 et DMEO en 58.

Christophe POULIN
et Marc GIRONDOT

PUISSANCE DES PC

Voici la réponse à la question de Marc Girondot.

La routine du PC-1500 qui interprète l'écriture des nombres commence à D14C pour la lecture d'une chaîne notée dans RAx.

La rencontre d'une lettre E envoie (par D1AD) à D21E, et UH est mis à 01 par D221. Dans la suite, si un point décimal est rencontré (D1AD puis D1FB), le test sur BO (UH) en D1FC évite la mise à 1 du B1, qui signalerait le point : ainsi celui-ci est négligé.

(On peut remarquer en passant qu'en l'absence de E, si un point a été enregistré, tout point lu ensuite sera négligé puisqu'il ne modifiera plus B1 : ainsi 17.18.19 se lit 17.1819).

Pour les chiffres suivant E, les zéros sont traités par D165 et les autres chiffres par D17B, envoyant toutes deux à D19E : ici le dernier chiffre lu est mis à la première place dans 7A00, celui qui y était prend la place du deuxième et ce dernier disparaît. On ne conserve donc bien que les deux derniers chiffres lus.

Encore une remarque : si la lettre E est répétée, elle n'est notée qu'une fois (voir D21F), de sorte que 1EEE2 se lit 100, 2E2E2 se lit 2E22, etc.

Dans tous les cas, les espaces sont négligés.

R. BOURION

SIMPLIFICATION DE FRACTIONS

Il faut simplifier des fractions en temps limité. L'ordinateur demande le niveau de difficulté de 1 à 5, le reste coule de source.

J.C. MOREL

```
4005:"B"CLEAR :RANDOM
4006:T1=TIME
4007:WAIT 0:PRINT "Niveau?(de_1_a_5):";:INPUT X:H=X*10:L=X*
     .02:CLS
4008:FOR K=0TO 9
4010:A=1+RND H:B=1+RND H:S=1+RND H
4013:IF A=BGOTO 4010
4015:C=ABS A:D=ABS B
4020:R=C-D*INT (C/D):IF R=0GOTO 4030
4025:C=D:D=R:GOTO 4020
4030:A=A/D:B=B/D
4040:WAIT 0:BEEP 3,70,10
4045:PRINT A*S;"/";B*S;"=";:INPUT "";B$
4050:IF B$=STR$ (A*S)+"/"+STR$ (B*S)THEN CLS :GOTO 4045
4080:A$=STR$ (A)+"/"+STR$ (B)
4083:IF INT (A/B)=A/BTHEN LET A$=STR$ (A/B)
4085:CLS :WAIT 300:IF B$<>A$THEN PRINT "Non!";A*S;"/";B*S;"
     =";A$:GOTO 4100
4090:N=N+1
4095:IF TIME -T1>=LTHEN LET K=10:BEEP 10,120,25:PRINT "APPR
     ENDS_A_COMPTER!!":GOTO 4110
4100:NEXT K
4103:IF N=10GOSUB 4130
4105:PRINT "Tu_as_obtenu:_";N;"/";K:IF N<9THEN PRINT "INSUF
     FISANT!"
4110:WAIT 100:PRINT "On_recommence?";"(O_ou_N):";:INPUT "";
     C$
4115:CLS :IF C$="O"THEN 4005
4125:CALL &CD71
4130:BEEP 1,117,291:RETURN
```

36-16
HG+ENVOI
BIENTÔT UNE RUBRIQUE
SHARPENTIER



Non, ce n'est pas encore la seconde partie, ce n'est pour l'instant que la suite du n° 1.

INITIATION LH-5801

Nous avons parlé dans l'initiation 1 du POKE, mais par inadvertance j'ai sauté le POKE à plusieurs paramètres. La syntaxe est la suivante :
POKE &AAAA,&VO,&V1,&V2,... &Vn.
Où &AAAA représente une adresse, soit un nombre de 0 à 65535 et &Vk une valeur de 0 à 255 donc sur un octet. L'effet sera le suivant :
&V0 sera mis en &AAAA
&V1 sera mis en &AAAA+1
&V2 sera mis en &AAAA+2
&Vn sera mis en &AAAA+n.
On s'en sert souvent pour taper de longs programmes, par exemple ceux publiés dans le SHARPENTIER :

Pour CLASSEMENT du n° 18 page 28 :

```
00C0:00 00 00 00 00 FD 1A FD : D4
00C8:1A 54 DE 25 D0 08 22 24 : 57
```

Ce genre de listing indique qu'à l'adresse &00C0 il y a la valeur &00, de même qu'en &00C1, &00C2, &00C3, &00C4, en &00C5 il y a la valeur &FD, en &00C6 la valeur &1A, en &00C7 la valeur &FD. La valeur &D4 placée à droite de cette ligne correspond à une valeur de contrôle qui ne sert qu'avec le moniteur SHARP publié dans le n° 9. On passe donc à la ligne suivante, et on voit qu'en &00C8 il faut mettre la valeur &1A.
Cela se traduira par :
POKE &00C0,&00,&00,&00,&00,&00,&FD,&1A,&FD,&1A
Passons à plus intéressant...

INITIATION LH-5801 (II)

Nous revoici donc ensemble pour une seconde partie où, comme promis, nous allons étudier à fond la MEV utilisateur.

Tous les chiffres seront écrits en hexadécimal avec entre parenthèses la conversion décimale. A partir du prochain article il n'y aura que la forme hexadécimale car nous entamerons alors l'étude du LM proprement dit et l'hexadécimal reflète mieux l'état des bits de la machine, puisque, soit un nombre HG en hexadécimal, G correspond directement aux bits 0 à 3 et H 4 à 7.

STRUCTURE DE LA MEV UTILISATEUR

La MEV utilisateur de votre PC s'étend quelque part de l'adresse 0000(0) à l'adresse 6FFF(28671). Suivant votre module mémoire et votre version de PC, plus ou moins de cette zone sera effectivement occupée. Il a été publié dans le n° 12 page 47 la carte complète de la mémoire suivant le module utilisé. Votre PC sait à tout moment quel est le module qu'il possède en regardant l'octet en MEV système de n° 7863 (30819). Celui-ci contient en effet le poids fort de l'adresse du premier octet de MEV utilisateur. Il convient de préciser ce qu'est exactement la notion de poids fort et de poids faible : Les adresses sont codées sur 16 bits numérotées de 15 à 0. La notation en hexadécimal : GHIJ reflète le mieux cet état :
J = bits 0 à 3
I = bits 4 à 7
H = bits 8 à 11
G = bits 12 à 15
On définit le poids fort comme étant la valeur GH, donc correspondant aux bits 8 à 15 donc les plus élevés. Le poids faible sera la valeur IJ correspondant aux bits 0 à 7 donc les plus faibles. Si on travaille sur des nombres décimaux, la conversion est moins évidente mais revient au même : soit N une adresse :
Poids fort de N = INT (N/256)
Poids faible de N = N AND 255

Nous savons à présent que la mémoire de nos PC commence en : PEEK (&7863)*256
en effet le poids faible de l'adresse du début de la mémoire est toujours 0, ceci par contraintes techniques.

Il nous reste à déterminer la fin de notre mémoire utilisateur. En fait, on déterminera plutôt le début de la mémoire qui n'est plus de la MEV utilisateur, et on saura que l'adresse précédente correspondra au dernier octet de la MEV utilisateur.

Le poids fort de cette adresse est donné par l'octet 7864(30820), le poids faible étant toujours 0. Donc le dernier octet de MEV utilisateur sera :
PEEK (&7864)*256—1

On appelera MM00—1 cette valeur et NN00 la valeur correspondant à PEEK (&7863)*256 donc au début de la MEV utilisateur.

La fois prochaine nous commencerons l'étude du codage du mode réserve.

Marc GIRONDOT

PLOT X,Y

Le PC 1260/61 n'a pas un écran graphique puisque composé de 48 matrices 5×7 différentes. Mais qu'importe, car à l'instar des autres PC comme le 1251, on peut adresser l'écran point par point par des POKE en BASIC ou par un programme en Langage Machine (1680 pts).

C'est le but de cet utilitaire en LM. Ne vous laissez pas tromper, le listage est du BASIC, mais en fait que de POKE !! Avant d'introduire le chargeur Basic, faite NEW ou All Reset, puis EQU# 1. Le LM va se placer dans la mémoire réservée au tableur. Vous ne pourrez plus vous servir de la fonction tableur, mais en revanche le LM sera protégé du Basic et ineffaçable par NEW. Une fois cela fait, entrez le Basic, vérifiez-le et taper RUN. L'exemple situé à la fin du chargeur vous permettra de contrôler le bon fonctionnement de la nouvelle fonction créée : il remplit toutes les matrices colonne par colonne.

Vous pouvez faire NEW et utiliser le programme LM dans vos œuvres Basic facilement, voici comment :
Il faut mettre le display On par un CALL &043B, puis mettre dans la variable X la coordonnée X désirée (entre 0 et 119), et dans Y la coordonnée Y désirée (entre 0 et 13). Le point choisi s'allumera instantanément par le CALL &4001. L'origine (0,0) est située en haut à gauche de l'écran.
L'adaptation au PC 1260 s'effectue sans problèmes car le programme LM est relogeable, il suffit simplement de changer l'adresse à chaque POKE comme suit : &4001 se change en &5801, &4011 en &5811... &4081 en &5881. En fait seul change le poids fort &40 en &58. Le CALL sera alors CALL &5801.
Il ne vous reste plus qu'à vous lancer dans les tracés de courbes et autres applications graphiques rendues facilement programmables.

Patrice VOISIN

```
  5:REM PLOT X,Y
  6:REM VOISIN PATRICE
 10:POKE &4001,&88,&10,&
    65,&09,&57,&11,&0A,&
    58,&66,&01,&38,&0A,&
    02,&0A,&DB,&57
 20:POKE &4011,&64,&0F,&
    44,&2C,&06,&57,&58,&
    64,&0F,&DB,&89,&10,&
    65,&11,&57,&58
 30:POKE &4021,&11,&12,&
    66,&02,&38,&14,&02,&
    64,&DB,&57,&64,&0F,&
    43,&34,&70,&0A
 40:POKE &4031,&2F,&03,&
    11,&13,&57,&58,&44,&
    2C,&18,&66,&01,&38,&
    11,&57,&64,&0F
 50:POKE &4041,&DB,&57,&
    58,&64,&0F,&43,&34,&
    70,&0A,&2F,&03,&2C,&
    04,&57,&58,&DB
 60:POKE &4051,&59,&67,&
    3C,&3A,&0C,&75,&3C,&
    43,&84,&DB,&50,&02,&
    28,&DB,&2C,&08
 70:POKE &4061,&43,&84,&
    DB,&50,&02,&20,&DB,&
    88,&59,&67,&07,&3A,&
    09,&75,&07,&DB
 80:POKE &4071,&84,&59,&
    74,&40,&DB,&04,&88,&
    02,&06,&DB,&45,&59,&
    34,&02,&80,&D2
 90:POKE &4081,&2F,&02,&
    DB,&57,&47,&53,&37
 95:REM * EXEMPLE *
100:CLS : CALL &43B: FOR
    X=0 TO 119: FOR Y=0
    TO 13: CALL &4001:
    NEXT Y: NEXT X: FOR
    I=0 TO 1000: NEXT I
```

LIST RSV

Dans le Sharpentier n° 9, nous avions présenté ce programme pour lister la zone réserve des PC 1251, il est cette fois adapté aux PC 1260/61.

Gageons que vous saurez l'adapter à votre PC (si il a un mode RSV) grâce aux renseignements donnés dans nos tableaux sur la mémoire système à partir du n° 14. Taper le Basic et RUN donnera sur écran le listage de la RSV, alors que PRINT=LPRINT suivi de GOTO «RSV» listera la RSV sur l'imprimante.

Le Club

```
1:"RSV" PRINT "RSV":
  FOR I=1 TO 48:X=
  PEEK (I+&65CF): IF (
  X>&F0 AND X<&FB) OR
  (X>&80 AND X<&8F)
  GOSUB 4: PRINT A$;B
2:IF X>0 NEXT I
3:PRINT " MEM "; STR$
  (49-I): END
4:R=X-64: IF X>&F0 LET
  R=R-96-49*(X=&F1)-23
  *(X=&F4)
5:POKE &65C8,&F5,R,&3A
  ,0:P= INT ((&65D0+I)
  /256):Q=&65D0+I-256*
  P
6:FOR J=1 TO 47:K=
  PEEK (&65CF+I+J): IF
  (K>&FA OR K<&F1) AND
  (K<&81 OR K>&8E)
  NEXT J
7:POKE &65C0,0,0,0,0,&
  D0,Q,P,J-1: RETURN


RSV
A: PEEK &65D0
 : LIST
=:^3*(4+25/4-2)
S: PRINT = LPRINT
X: GOTO "RSV"@
G:?&%$#"!π√,<>
 MEM 0
```





THE ASTEROID ADVENTURE

Ce jeu galactique est extrait d'une revue japonaise I/O et fonctionne sur les PC 1251/55. Votre mission est cette fois-ci de sillonner l'espace à travers un nuage d'astéroïdes destructeurs.

Pour ce faire, tapez le programme Basic fourni. Certaines lignes saturent les 80 octets autorisés ; entrez-les en plusieurs fois, en validant tour à tour par ENTER et en éditant avec la flèche de curseur droit. Vérifiez bien les DATA et lancez le jeu par RUN. Une fenêtre apparaît, vous êtes à droite et un nuage d'astéroïdes fonce droit sur vous. Vous pouvez éviter les obstacles avec les touches 8 et 2, respectivement Haut et Bas par paliers successifs. Vous perdez lorsqu'un astéroïde vous touche ; c'est le Crash spectaculaire. A ce moment le Score apparaît et on repart par un appui sur ENTER.

Les POKE sont utilisés pour le graphisme à l'écran et le programme reste donc en Basic : une adaptation à d'autres PC est possible en connaissant la correspondance du CALL &11E0 (cf tableaux s/pgms) et les adresses de la ram vidéo (cf N° 12).
Galactiquement vôtre.

JFV

```
10:CLEAR :Y=4: DIM B(7)
   :B(1)=1:B(2)=2:B(3)=
   4:B(4)=8:B(5)=16:B(6
   )=32:B(7)=64
15:POKE &F83A,65,127:
   POKE &F821,127,65
20:FOR C=0 TO 1: CALL &
   11E0:A$= INKEY$ : IF
   A$ GOTO 40
30:NEXT C: GOTO 70
40:IF A$="8" LET Y=Y-(Y
   >1): GOTO 60
50:IF A$="2" LET Y=Y+(Y
   <7)
60:POKE &F836,B(Y)+X,B(
   Y)+Z,B(Y-1)+B(Y+(Y<7
   )): GOSUB 120: NEXT
   C
```

```
 70:READ D: IF D=-1
    RESTORE 200:D=0
 80:Z=X:X=W:W=U:U=T:T=S:
    S=R:R=Q:Q=P:P=O:O=N:
    N=M:M=L:L=K:K=J:J=I:
    I=H:H=G:G=F:F=E
 90:E=D
100:POKE &F823,D,E,F,G,H
    ,I,J,K,L,M,N,O,P,Q,R
    ,S,T,U,W,X+B(Y),Z+B(
    Y),B(Y-1)+B(Y+(Y<7))
110:GOSUB 120:V=V+10:
    GOTO 20
120:IF ((X AND B(Y))<>0)
    OR ((Z AND B(Y))<>0)
    GOTO 140
130:RETURN
140:WAIT 0: PRINT "
      ": FOR I=0 TO 3:
    CALL &11E0: POKE &F8
    28,0,0,0,0,0,0,20,32
    ,2,20,0,0,0,0,0
150:POKE &F828,0,0,0,0,0
    ,8,82,0,33,66,0,37,8
    ,0,0,0,0
160:POKE &F828,32,4,40,6
    6,33,0,64,2,32,65,16
    ,36,9,2: NEXT I
180:WAIT : PRINT "SCORE
    : "; STR$ V
190:WAIT 0: PRINT "
              ": RESTORE
    200: GOTO 10
200:DATA 8,28,8,65,99,65
    ,8,28,8,0,34,85,34,0
    ,8,20,8,64,34,37,66,
    34,37,66,0,16,54,9
210:DATA 31,9,54,16,0,0,
    0,4,72,48,120,48,72,
    4,0,0,0,0,33,27,37,3
    7,27,33,0,0,34,85
220:DATA 85,34,0,0,31,2,
    5,114,114,5,2,31,0,0
    ,0,124,0,0,0,31,0,0,
    0,96,0,0,63,0,0,32,0
230:DATA 68,52,85,119,85
    ,52,68,0,32,0,0,0,0,
    28,8,8,28,0,0,0,28,0
    ,3,112,0,0,111,0,0,0
240:DATA 124,0,0,0,111,0
    ,0,0,125,0,0,0,31,4,
    10,4,31,0,0,0,120,72
    ,72,120,0,0,15,9,9
250:DATA 15,0,0,120,72,7
    2,120,0,0,15,9,9,15,
    0,0,119,85,119,65,65
    ,8,20,4,20,8,65,34
260:DATA 20,17,17,29,17,
    17,16,32,64,0,0,0,64
    ,42,42,42,0,0,0,14,1
    8,35,2,6,0,0,56,74
270:DATA 74,74,8,0,124,1
    20,96,67,15,31,63,31
    ,15,67,96,120,124,12
    0,96,67,111,67,0,24
280:DATA 61,61,24,0,64,1
    12,28,70,96,16,9,11,
    9,16,96,70,28,112,64
    ,-1
```

POUR TOUTE CORRESPONDANCE AVEC LE CLUB JOINDRE UNE ENVELOPPE TIMBRÉE



INVERSE PARTIEL

Comme son nom l'indique, ce programme LM permet des inversions vidéo (blanc sur noir) d'une partie de l'écran graphique de votre PC 1350.

Plus particulièrement, il permet d'inverser ou de souligner un texte. Non relogeable, le programme LM sous forme d'un DUMP se rentre en mémoire avec l'utilitaire fourni dans notre numéro 13.

Ensuite l'utilisation est simple. Exemple :
10 : WAIT 0 : CLS
20 :PRINT "INVERSE PARTIEL" : CALL &6800
Inverse sur l'écran la phrase INVERSE PARTIEL.

Si l'on remplace CALL &6800 par CALL &680D cette phrase INVERSE PARTIEL sera uniquement soulignée. Essayez pour vous en convaincre.

Le texte à inverser peut également être placé à n'importe quel endroit de l'écran. Exemple :
10 : WAIT 0 : CLS
20 : CURSOR 16,1 : PRINT "INVERSE PARTIEL"; : CALL &680D : PRINT " PAR "; : PRINT "CAPLA" : CALL &6800.

PRÉCAUTIONS D'EMPLOI

• Dans la fin de la ligne 20 exemple 2, si on remplace PRINT " PAR ";: PRINT "CAPLA": CALL &6800 par PRINT " PAR ";" CAPLA ": CALL &6800, c'est toute la phrase PAR CAPLA qui se retrouve inversée.

• Lorsque l'on a inversé quelque chose à l'écran, ne jamais faire suivre d'un PRINT sans argument. Exemple, si on place à la suite d'un des deux programmes précédents la ligne :
30 : PRINT
l'écran reste comme si aucun changement n'avait été effectué. Il faudra plutôt mettre :
30 : PRINT ""

Vous pouvez maintenant agrémenter vos programmes BASIC d'une présentation professionnelle.

B. CAPLA

```
6800 : 10 68 BE 02 FF 52 11 C0 <5A>
6808 : 02 45 52 2C 0C 10 68 BE <07>
6810 : 02 80 52 11 C0 02 44 52 <3D>
6818 : 03 78 02 8B E2 97 24 8A <2F>
6820 : DB 59 80 DB 78 68 C9 34 <6C>
6828 : 24 8B DB 59 80 DB 5B 78 <11>
6830 : 68 E4 E2 B5 03 6D 02 60 <B5>
6838 : E2 97 24 67 00 38 86 8A <4C>
6840 : 71 01 63 FF 28 06 70 18 <8A>
6848 : 8B 71 01 24 67 00 38 03 <C3>
6850 : 2D 12 8A 59 80 DB 78 68 <5D>
6858 : C9 34 8B 59 80 DB 5B 78 <0F>
6860 : 68 F8 E2 97 06 84 59 DA <96>
6868 : 86 59 83 45 63 00 28 0C <3E>
6870 : 87 59 DA 85 59 83 45 63 <C3>
6878 : 00 38 4B 84 63 7B 38 0F <2C>
6880 : 63 5D 38 0B 63 3B 38 07 <E0>
6888 : 63 1D 38 03 2C 23 71 1D <98>
6890 : 85 70 02 63 7A 28 30 71 <9D>
6898 : 0A 84 63 00 28 05 70 44 <D2>
68A0 : 2C 0F 63 40 28 05 71 1E <9A>
68A8 : 2C 07 63 1E 28 03 70 44 <93>
68B0 : 24 67 FF 38 0F 67 80 28 <E0>
68B8 : 05 00 00 80 DB 00 FF 80 <DF>
68C0 : 45 DB 52 2D 5F 37 05 2D <67>
68C8 : 18 02 70 80 63 05 3A 07 <B3>
68D0 : 71 05 74 02 2D 09 DA 02 <FE>
68D8 : 05 63 00 38 07 74 06 71 <92>
68E0 : 01 2D 09 37 63 00 38 28 <31>
68E8 : 67 05 28 0D 83 63 70 28 <1F>
68F0 : 08 02 1D 03 78 80 71 01 <94>
68F8 : 80 63 03 28 05 74 5E 2C <11>
6900 : 0F 63 02 28 05 74 1E 2C <5F>
6908 : 07 63 01 28 03 74 40 37 <81>
```

DEZHEX

Cette petite routine en langage machine offre une conversion décimale-hexadécimale qui est désormais classique, mais avec un confort accru.

En effet, ce programme permet d'affecter le résultat à une variable de caractère de votre choix. Et ce, sous Basic avec une gestion d'un pseudo mot clef, par passage des paramètres après un CALL (cf n° 16, p. 44). Le programme DEMO DEZHEX montre le fonctionnement très simple. Il faut bien sûr rentrer le programme LM donné, sous forme de DUMP avec l'utilitaire du n° 13, ou par la fonction POKE xxyy,zz,tt, etc..., maintes fois décrite.

ENSUITE LA SYNTAXE SERA :

CALL &6000, variable $ = (expression num)
avec 0< = expression num < = 65535, où l'expression peut être le résultat d'un calcul.
La gestion des erreurs est comme pour le Basic :
ERROR 1 à 9 : comme SHARP
ERROR 0 : erreur de syntaxe.

EXEMPLE :

CALL &6000,Z$ = 255 donne Z$ = "FF"

A = 10:B = 20
CALL &6000,HE$ = (A + B) donne HE$ = "1E"

Pour les amateurs d'adaptation, le programme n'est pas relogeable, et la correspondance des CALL en ROM interne et adresses système utilisés sera donnée dès que connue. Une astuce permet d'obtenir un résultat différent dans un sous-programme, en changeant l'adresse d'appel ; le code de l'instruction étant camouflé comme opérande du mnémonique WAIT. Exemple, en &6049 on a WAIT &DA si on fait CALL ou JP &6049, mais on a EXAB si on fait JP &604A (code &DA camouflé)

Norbert ENNULAT
et Klaus DITZE

```
DUMP DEZHEX
-----------
6000 24672C380723F0E6:EF
6008 5B5B37B3609F6102:02
6010 FA543B0BDAB36204:87
6018 0209391523629028:96
6020 02DA106F26521124:08
6028 861B24673D292978:33
6030 80393B2BF63A3B2F:B9
6038 106F24861A112657:D1
6040 6700280702F52603:B6
6048 074EDA995978605D:56
6050 985978605D23C33A:46
6058 0226D12D54345878:7E
6060 60635B640F670A3A:3C
6068 0374077430C33A03:22
6070 264EC23700000000:6D

 10:REM DEMO DEZHEX
 20:CLS : CLEAR
 30:INPUT "DEZ: ";D
 40:IF D<0 END
 50:CALL &6000,H$=D
 60:PRINT "HEX: ";H$
 70:GOTO 30
```

```
DEZHEX
6000 24     IXL
6001 672C   CPIA 2C
6003 3807   JRZP 600B
6005 23     CLA
6006 F0E6   CAL  10E6
6008 5B     POP
6009 5B     POP
600A 37     RTN
600B B3     LP 33
600C 609F   ANIM 9F
600E 6102   ORIM 02
6010 FA54   CAL  1A54
6012 3B0B   JRCM 6008
6014 DA     EXAB
6015 B3     LP 33
6016 6204   TSIM 04
6018 0209   LIA 09
601A 3915   JRZM 6006
601C 23     CLA
601D 6290   TSIM 90
601F 2802   JRNZP 6022
6021 DA     EXAB
6022 106F26 LIDP 6F26
6025 52     STD
6026 1124   LIDL 24
6028 86     LP 06
6029 1B     EXBD
602A 24     IXL
602B 673D   CPIA 3D
602D 2929   JRNZM 6005
602F 788039 CALL 8039
6032 3B2B   JRCM 6008
6034 F63A   CAL  163A
6036 3B2F   JRCM 6008
6038 106F24 LIDP 6F24
603B 86     LP 06
603C 1A     MVBD
603D 1126   LIDL 26
603F 57     LDD
6040 6700   CPIA 00
6042 2807   JRNZP 604A
6044 02F5   LIA F5
6046 26     IYS
6047 0307   LIB 07
6049 4EDA   WAIT DA
604B 99     LP 19
604C 59     LDM
604D 78605D CALL 605D
6050 98     LP 18
6051 59     LDM
6052 78605D CALL 605D
6055 23     CLA
6056 C3     DECB
6057 3A02   JRCP 605A
6059 26     IYS
605A D1     RC
605B 2D54   JRM 6008
605D 34     PUSH
605E 58     SWP
605F 786063 CALL 6063
6062 5B     POP
6063 640F   ANIA 0F
6065 670A   CPIA 0A
6067 3A03   JRCP 606B
6069 7407   ADIA 07
606B 7430   ADIA 30
606D C3     DECB
606E 3A03   JRCP 6072
6070 26     IYS
6071 4EC2   WAIT C2
6073 37     RTN
```


IL ÉTAIT UNE FOIS, UN PC DANS L'OUEST

Vous l'avez deviné, ce jeu retrace une des plus célèbres scènes de western, à savoir le duel final ! Vous allez voir s'affronter en direct sur l'écran de votre PC deux cow-boys rivaux : vous à gauche et le PC à droite.

MODE D'EMPLOI :

Une fois le compte à rebours terminé, il vous faut guider la trajectoire de votre balle à travers le décor (2 cactus plus une échelle), afin de blesser votre adversaire qui mourra à la suite de 6 blessures. Attention ! la difficulté intervient aux deux-tiers du parcours, lorsqu'un mur s'érige devant vous, protégeant ainsi votre cible. Vous devez réussir à le franchir en passant par l'orifice qui évolue régulièrement dans le sens vertical. Si vous réussissez, une croix remplace l'adversaire et votre score augmente de 1 automatiquement. Par contre, si votre balle heurte une partie du décor, c'est vous qui perdez.

QUELQUES PRÉCISIONS :

- Il y a deux niveaux de difficultés influant sur la largeur du mur.
- La touche 7 permet de guider la balle vers le haut et la touche 1 vers le bas.
- La partie s'arrête à 5, le PC désigne alors le vainqueur puis une nouvelle partie commence.
- Le jeu se lance par DEF A.
- Le programme fait 1843 octets.
- Le décor s'efface au fur et à mesure de la progression de votre projectile, ne laissant apparaître que sa trajectoire.
- Ne vous inquiétez pas si des parasites s'agitent sur les bords de l'écran : c'est normal et dû au graphisme.

Voilà, l'essentiel a été dit. Il ne me reste qu'à vous souhaiter un bon amusement !

Olivier SCHANEN

```
    5:REM *OLIVIER SCHANEN
       *
   10:"A":INPUT "NIVEAU 1-
       2 ? ";Y:IF Y<>1 AND
       Y<>2 THEN 10
   20:RESTORE 1050:Z=0:V=0
       :GOTO 150
   30:IF V=3 OR Z=3
       RESTORE 1100
   35:IF Z=5 POKE &6045,0,
       0,0,0,0,62,65,65,65,
       62,7,24,96,24,7,127,
       9,9,9,127:GOTO 2500
   40:IF V=5 POKE &6045,0,
       0,0,0,0,62,65,65,65,
       62,7,24,96,24,7:GOTO
       2700
   50:IF Z=4 POKE &6023,12
       7,97,111,71,127
   60:IF V=4 POKE &6040,12
       7,71,111,97,127
   70:IF Z=3 POKE &6023,12
       7,85,85,65,127
   80:IF V=3 POKE &6040,12
       7,65,85,85,127
   90:IF Z=2 POKE &6023,12
       7,69,85,81,127
  100:IF V=2 POKE &6040,12
       7,81,85,69,127
  110:IF Z=1 POKE &6023,12
       7,123,65,127,127
  120:IF V=1 POKE &6040,12
       7,127,65,123,127
  130:IF Z=0 POKE &6023,12
       7,65,93,65,127
  140:IF V=0 POKE &6040,12
       7,65,93,65,127
  145:FOR X=1 TO 300:NEXT
       X
  150:WAIT 0:PRINT "":CALL
       &5A2
  200:POKE &6023,80,106,31
       ,106,80
  210:POKE &6040,8,106,31,
       106,8
  230:POKE &605E,24,16,126
       ,16,28
  240:POKE &604F,28,16,126
       ,16,24
  250:POKE &6059,127,42,12
       7:FOR X=1 TO 60:NEXT
       X
  260:POKE &6059,63,21,63:
       FOR X=1 TO 60:NEXT X
  270:POKE &6059,31,10,31:
       FOR X=1 TO 60:NEXT X
  280:POKE &6014,127,85,85
       ,65,127:FOR X=1 TO 5
       0:NEXT X
  290:POKE &6014,127,69,85
       ,81,127:FOR X=1 TO 6
       0:NEXT X
  300:POKE &6014,127,123,6
       5,127,127:FOR X=1 TO
       60:NEXT X
  360:POKE &6023,106,31,42
       ,68,4
  400:A=&6067:B=4
  500:A=A-1:N=B>1:M=B<64
  520:IF A=&6044 POKE &604
       0,127,123,97,123,127
       :FOR X=1 TO 500:NEXT
       X:Z=Z+1:GOTO 30
  540:IF (B AND (PEEK A))<
       >0 POKE &6023,127,12
       3,97,123,127:FOR X=1
       TO 500:NEXT X:V=V+1:
       GOTO 30
  542:IF Y=2 THEN 550
  545:IF A<&6053 READ D:
       POKE &6047,D:GOTO 56
       0
  550:IF A<&6053 READ D:
       POKE &6047,D:POKE &6
       048,D
  560:POKE A,B
  600:E$=INKEY$
  700:IF E$="7" THEN LET B
       =B/(N+1):GOTO 500
  800:IF E$="1" THEN LET B
       =B*(M+1):GOTO 500
  900:GOTO 500
 1050:DATA 125,125,125,1
       23,123,119,119,119
       ,111,111,111,95,95
       ,95,125,125,125
 1100:DATA 123,123,123,1
       19,119,111,111,111
       ,95,95,95
 1150:DATA 125,125,125,1
       23,123,123,119,119
       ,119,111,111,111,9
       5,95,95
 1200:DATA 125,125,125,1
       23,123,123,119,119
       ,119,111,111,111,9
       5,95,95,125,125,12
       5
 2500:POKE &6059,79,41,2
       5,9,127,54,73,73,7
       3,127,0,0,0,0,0:
       FOR X=1 TO 40:NEXT
       X
 2510:POKE &6023,68,106,
       31,106,68:FOR X=1
       TO 50:NEXT X
 2520:POKE &6023,80,106,
       31,106,80:FOR X=1
       TO 50:NEXT X
 2530:POKE &6023,68,106,
       31,106,68:FOR X=1
       TO 500:NEXT X:GOTO
       10
 2700:POKE &6054,127,9,9
       ,9,127,79,41,25,9,
       127,54,73,73,73,12
       7,0,0,0,0,0,0,0,0,
       0,0
 2800:POKE &6040,68,42,3
       1,42,68:FOR X=1 TO
       50:NEXT X
 2900:POKE &6040,16,106,
       31,106,16:FOR X=1
       TO 50:NEXT X
 3000:POKE &6040,68,42,3
       1,42,68:FOR X=1 TO
       500:NEXT X:GOTO 10
```

INITIATION AU LM ESR-H (6e)

Pour vous permettre de comprendre davantage les programmes LM ou d'en construire à votre tour, nous allons étudier en priorité une gamme d'instructions très importante : INC et DEC. Nous reviendrons aux Moves et aux Échanges évolués la prochaine fois.

Incrémenter ou décrémenter un registre signifie simplement ajouter 1 à la valeur qu'il contient, respectivement retrancher 1. C'est-à-dire pour l'Accu, s'il contient la valeur &4F (79 en déc), en l'incrémentant A contiendra &50 (80 en déc). De même, si on décrémente A, la valeur contenue sera &4E (78 en déc). Voilà pour le principe, nous verrons par la suite les instructions agissant comme indiqué. Avant de compliquer les choses, rappelons que nous avons plusieurs sortes de registres ; notamment les registres d'index de la ram interne (par ex P) sur 7 bits, et les registres constituant cette ram interne (par ex A) sur 8 bits. Nous allons constater que nous ne pouvons pas incrémenter ou décrémenter tous les registres connus. Regardons du côté des registres d'index. Le plus usité est P, alors que Q et R ont chacun un rôle spécifique. En fait seul P a la possibilité de changer de valeur par une instruction INC ou DEC :

INCP : code &50 P+1 → P
DECP : code &51 P—1 → P

Comme P est sur 7 bits, sa valeur max est &7F (127 d). Si P vaut &7F et que l'on exécute INCP alors P vaudra 0, car on n'a pas de 8e bit pour passer à la valeur &80 (128 d). De même si P vaut 0, DECP donnera &7F. De toutes façons, la ram va de 0 à &5F et donc les valeurs comprises entre &60 et &7F n'ont pas lieu d'être utilisées, car sans significations réelles. INCP et DECP n'influent sur aucuns Flags ou autres registres, nous allons voir que ce n'est plus le cas pour les registres sur 8 bits. Pour ce faire, il faut maintenant étudier la structure d'un octet, son comportement quand on ajoute ou retranche 1 et donc l'influence sur les flags. Les flags C et Z fonctionnent en fait très simplement. C indique le dépassement dans un sens ou dans l'autre (voir fig. 1). Alors que Z indique le passage à la valeur 0. On se préoccupe des flags aux valeurs critiques suivantes : 0, &FF et 1. En effet, sur 8 bits, ajouter 1 à &FF (255) nous fait retourner à 0, et inversement enlever 1 à 0 donne &FF (255). Regardons en binaire comment cela s'écrit :

La valeur &FF se note 1111 1111.
Si on rajoute 1, on obtient 256 soit &100 ou en base 2 : 1 0000 0000.
Il nous faut donc 9 bits pour coder 256, or nous en avons 8. Seuls restent donc les bits 0 à 7. Le bit 8 qui n'existe pas est remplacé par le Flag C ; c'est lui le 9e bit ou qui en tient lieu. On l'appelle alors retenue ou en anglais Carry. De plus comme nous obtenons la valeur 0, le flag Z est mis lui aussi.

En résumé :
&FF + 1 → &00, C=1, Z=1.
De même manière pour DEC :
&00 — 1 → &FF, C=1, Z=0.
Nous admettrons ce résultat pour l'instant. Lors de l'étude des additions et soustractions, nous verrons en détail la représentation des nombres négatifs (en complément à 1 ou à 2 par ex.), ce n'est pas ici primordial. Pour information 1 est codé 0000 0001 en binaire. Le complément à 1 consiste à inverser les 0 avec les 1. —1 sera alors représenté par :
1111 1110.
Par ex. : 5 = (0000 0101)2.
En complément à 1 :
—5 = (1111 1010)2.
Le complément à 2 s'obtient simplement en ajoutant 1 à la première forme. Alors —1 deviendra :
1111 1111
(C'est le &FF de &00—1 avec C=1) et —5 = (1111 1011)2.
Il est possible aussi sur 8 bits d'utiliser le bit 7 pour indiquer un nombre négatif ; c'est le binaire signé, on va de —127 à +127. Le plus souvent nous utiliserons la représentation en complément à 2 pour sa facilité d'emploi. Cela permet avec une simple addition d'obtenir le bon résultat, avec dans le carry C la retenue négative. Nous y reviendrons une prochaine fois. Nous constaterons que les flags Z et C se comportent avec les mnémoniques du type AD ou SB, comme je l'ai décrit en général et pour INC et DEC. Le troisième cas se présente quand on décrémente 1 de la valeur 1, alors le Flag Z est mis :
1 — 1 → 0, C=0, Z=1.
Revenons maintenant à nos registres. Nous nous contentons d'ajouter ou de retrancher 1 avec les mnémoniques INCr et DECr où r est un registre 8 bits parmi : I, J, A, B, K, L, M, N. Voici donc les instructions correspondantes :

INCI	: code &40	I+1 → I
INCJ	: code &C0	J+1 → J
INCA	: code &42	A+1 → A
INCB	: code &C2	B+1 → B
INCK	: code &48	K+1 → K
INCL	: code &C8	L+1 → L
INCM	: code &4A	M+1 → M
INCN	: code &CA	N+1 → N
DECI	: code &41	I—1 → I
DECJ	: code &C1	J—1 → J
DECA	: code &43	A—1 → A
DECB	: code &C3	B—1 → B
DECK	: code &49	K—1 → K
DECL	: code &C9	L—1 → L
DECM	: code &4B	M—1 → M
DECN	: code &CB	N—1 → N

Vous noterez la symétrie des codes pour passer d'un registre à un autre dans un couple et de INCr à DECr. Cette fois-ci les flags sont affectés dans les cas déjà vus. De plus le registre Q est lui aussi modifié de manière interne ; vraisemblablement pour indexer les registres utilisés. Alors prudence quant à l'utilisation de Q, avant ou après l'une de ces instructions.

Pour finir, quelques exemples :

LIA &00 INCA donne &01 → A, C=0, Z=0
LIA &00 DECA donne &FF → A, C=1, Z=0
LIA &FF INCA donne &00 → A, C=1, Z=1
LIA &FF DECA donne &FE → A, C=0, Z=0
LIA &01 INCA donne &02 → A, C=0, Z=0
LIA &01 DECA donne &00 → A, C=0, Z=1
Les cas les plus spéciaux sont ici résumés.

Nous avons abordé l'incrémentation et décrémentation des registres 8 bits, il s'avère que ce type d'instructions existe pour les registres 16 bits X et Y : IX, IY, DX, DY, et aussi IXL, DXL et IYS, DYS avec des particularités importantes. Ces mnémoniques feront l'objet d'une leçon particulière. La fois prochaine nous reviendrons si vous êtes sages, moi en forme et le bulletin plus épais, sur les Moves et Echanges simultanés de registres contigüs.
A la leçon n° 7 !

Jean-François VIGNAUD

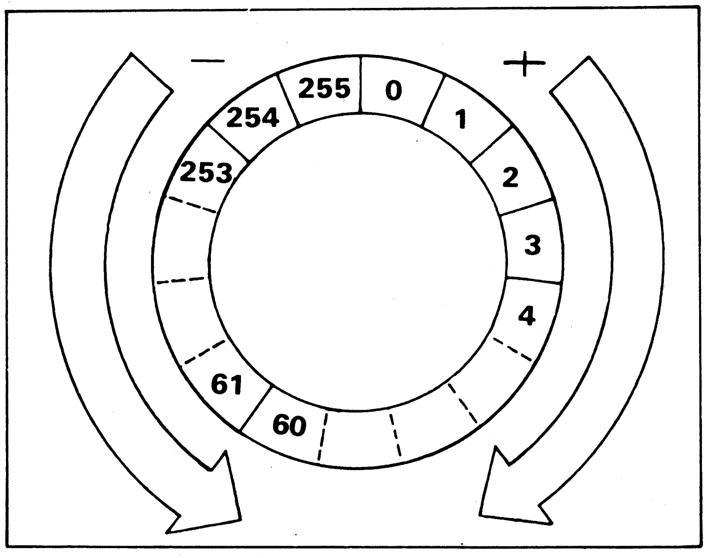


FIG. 1

PRÉCISION SUR LES DATES

Au fur et à mesure des réponses au petit concours du SHARPENTIER, des précisions nous ont été apportées. Ainsi M. BURGERT nous a signalé que les dates exactes qui n'existent pas sont :
Italie : 5/10/1582 au 14/10/1582
France : 10/12/1582 au 19/12/1582

D'ailleurs, à ce ce propos, que s'est-il passé dans les autres pays ? Par exemple en AMÉRIQUE où les nouvelles ne pouvaient arriver, si elles arrivaient, qu'avec plusieurs mois de retard ? Y-a-t-il des lecteurs qui se sentent une âme d'historien pour débrouiller ce mystère ? Je peux tout de suite vous dire que cette particularité n'est pas citée dans les livres « L'histoire de la FRANCE et des FRANÇAIS au jour le jour ». On aurait pourtant pu espérer, vu le titre que cela soit au moins commenté pour la FRANCE.
Nous avons de plus eu une autre précision de la part de M. ANDRÉ au sujet de l'an zéro. A-t-il existé ou non ? Et bien tout dépend de votre passion. Selon que vous êtes historien ou astronome, la réponse est différente. Les uns considèrent qu'avant l'an I il y avait l'an I avant J.C., et les autres considèrent que c'était l'an 0, certainement pour faciliter les calculs de position de planètes. Une autre question : on sait que dans la mythologie, J.C. est né le 25 décembre, mais en quelle année ? On pourrait imaginer que c'est en l'an 0 ce qui correspond en l'an −1 avant J.C., donc avant sa naissance. Bizarre ! Mais où cela devient comique, c'est qu'en fait J.C. serait né en 4 ou 5 avant J.C., son année de naissance ayant été désignée au VI[e] siècle par DENYS LE PETIT sans preuve. De plus cela ne serait pas le 25/12, mais au printemps !

Voici de plus quelques compléments historiques sur notre calendrier :
Les Égyptiens donnaient 365 jours à une année. En 46 avant J.C., Jules César introduit l'année bissextile tous les 4 ans. Donc l'année du calendrier JULIEN avait 365,25 jours. La seconde réforme fut celle du pape GRÉGOIRE XIII qui définit le calendrier GRÉGORIEN sans année bissextile la dernière année d'un siècle sauf tous les 4 siècles. L'année a donc 365,2425 jours.

Voici ce que cela donne en informatique :
Bissextile
INT (A/100)/4 = INT (INT (A/100)/4)
Non bissextile
A/100 = INT (A/100)
Bissextile
A/4 = INT (A/4)

Les astronomes modernes donnent 365.2422 jours à une année, et ce sont donc les MAYAS qui s'en rapprochent le plus avec 365.2420 jours par an. On peut de plus remarquer qu'être informaticien chez les Égyptiens aurait été un rêve, puisque tous leurs mois avaient le même nombre de jours.

De plus voici une simplification par M. NOUAIL d'une des routines publiées dans le n° 17 :

N = Nombre de jours d'un mois M d'une année A :
N = ASC
MID$("&#&%&%&&%&%&",M,1) − 7 + (A/2 = INT (A/4)*M)

On gagne un octet par rapport à la version précédente. On pourrait gagner un peu plus en POKANT dans la chaîne à la place de & le code 31, à la place de % le code 30 et 28 en remplacement de #, et dans ce cas là on supprime le −7, d'où gain d'encore 2 octets.

Il ne reste plus qu'à faire une conversion de notre calendrier dans les autres, par exemple le calendrier révolutionnaire, et nous attendons donc vos propositions d'articles.
A bientôt.

1986 après J.C.
M. Jacques ANDRÉ
M. Albert BURGERT
M. Marc GIRONDOT
M. Nicolas NOUAIL

RÉSULTATS DU CONCOURS N° 17

Tout d'abord bravo à tous, nous avons reçu assez de réponses pour être confronté au délicat problème du classement. D'autant plus que nous avons reçu autant de résolutions différentes que de réponses, illustrant le célèbre proverbe informatique :
« Il y a autant de solutions à un problème que de programmeurs capables de le résoudre ».

En fait, les plus grosses divergences portaient sur les limites de validité du programme. Il y a une très grande diversité dans les réponses :
Certains tiennent compte des années négatives, d'autres commencent en 1/1/1, certains le 15/10/1582 et d'autres enfin le 1/3/1900 comme d'ailleurs tous les programmes de l'article du n° 17.

Vous conviendrez donc qu'il était très difficile de faire une sélection, et c'est pourquoi nous avons choisi de ne pas en faire. Vous trouverez tous les programmes qui nous ont été envoyés. En fait, 14/7/1789 + 70000 donne le 10/3/1981. On trouvait le 8/3/1981 avec les programmes qui ne marchent que jusqu'au 1/3/1900. Les tests de rapidité ont été effectués sur PC-1500 avec ce programme :

```
1 : WAIT 0 : TIME = O : FOR I = OTO 49
  : J = 14 : M = 7 : A = 1789 : N = 70000
  : GOSUB 10 : PRINT J;M;A : NEXT I
  : WAIT : PRINT TIME
```

Sachant qu'il met 4 secondes pour s'exécuter à vide. La longueur du programme a aussi été calculée sur PC-1500, où un mot-clé prend deux octets. J'ai parfois modifié le listing original pour qu'il réponde exactement à la question.

Je pense que personne ne pourra contester la première place à M. ANDRÉ qui a fait un programme utilisant seulement 2 variables B et C, relativement court, 326 octets et allant vite, 0,64 secondes. Ce qui fait sa grande force, c'est qu'il fonctionne à partir du 1.1.1

Nous mettrons tous les autres programmes ex-aequo tant il est difficile de faire un classement. Toutefois, nous féliciterons principalement Nicolas NOUAIL qui a fait le programme le plus court, 174 octets, et utilisant le moins de variables, 0. Toujours dans les félicitations, M. TOUSSAINT est à citer car c'est son programme qui est le plus rapide sur le test : 0,42 seconde.

De même M. DALLET est le meilleur pour ce qui est du domaine de travail puisqu'il travaille aussi sur les années négatives en sautant l'année 0.

Voici le tableau complet :

nom	octets	domaine	nb var.	temps
ANDRÉ	326	1/1/1	2	0.64
NOUAIL	174	1/3/1900	0	0.6
TOUSSAINT	201	1/3/1900	1	0.42
DALLET	518	Infini	7	0.6
BURGERT	405	15/10/1582	4	1.58
DOIDY	388	1/3/1900	3	23

Bravo à tous...

Le Club

Mr ANDRE

```
950:B=A+M/100+J/1E4-1582.1005,C=M<3,M=C*12+M+1,A=A-C
955:A=(B>=0)*(INT (INT (A/100)/4)-INT (A/100))+INT (365.25
    *A)-2*(B<0)
960:A=A+INT (30.6*M)+J+N,J=INT (A/36524.25-4)
965:B=A+1722521+(A>578163)*(1+J-INT (J/4)),C=INT (B/365.25
    -.3343)
970:M=INT (365.25*C),N=INT ((B-M)/30.6001),J=B-M-INT (30.6
    001*N),M=N-1-12*(N>13.5)
975:A=C-4716+(M<2.5):RETURN
```

Mr NOUAIL

```
2:N=INT (365.25*(A-(M<3)))+J+INT (30.6*(M+1+12*(M<3)))+N
   -63,A=INT (N/365.25),M=1
3:J=INT (N+1-A*365.25)
4:N=ASC MID$ ("&#&%&%&&%&%&",M,1)-7+(A/2=INT (A/4)*M):
   IF N<JLET M=M+1:J=J-N:GOTO 4
5:RETURN
```

Mr TOUSSAINT

```
10:N=INT (365.25*(A-(M<3)))+J+INT (30.6*(M+1+12*(M<3)))+N
20:A=INT (N/365.25):P=N-INT (A*365.25):M=INT (P/(365+(A/4
   =INT (A/4)))*10)
30:J=N-INT (30.6*(M+1+12*(M<3)))-INT (365.25*(A-(J=0)))
40:J=J+29*(J=0):M=M-(J=0):RETURN
```

Mr DALLET

```
160:D=J+100*M+1E4*A
170:IF A=0GOTO "JP"
180:IF A<0LET A=A+1
190:IF M<3LET A=A-1:M=M+12
200:IF D>=15821015GOTO "GREG"
210:IF D<15821005GOTO "JUL"
220:"JP"PRINT "Jour_n_existant_Pas...":END
230:"GREG"V=INT (A/100):W=2-V+INT (V/4)
240:"JUL"J=INT (365.25*A)+INT (30.6001*(M+1))+J+W+1720995
250:J=J+N
260:IF J<2299161LET T=J:GOTO "T1"
270:R=INT ((J-1867216.25)/36524.25)
280:T=J+1+R-INT (R/4)
290:"T1"U=T+1524
300:V=INT ((U-122.1)/365.25)
310:W=INT (365.25*V)
320:X=INT ((U-W)/30.6001)
330:J=U-W-INT (30.6001*X)
340:M=X-1:IF X>13.5LET M=X-13
350:A=V-4716:IF M<2.5LET A=V-4715
360:IF A<1LET A=A-1
370:RETURN
```

Mr BURGERT

```
200:GOSUB 510:P=I:N=N+P
400:B=365.2426:A=INT (N/B):M=INT ((N-A*B)/30.6):J=INT (N-A
    *B-M*30.6)
410:A=A+1:M=M+1:GOSUB 600
420:B=28+VAL MID$ ("3"+STR$ F+"3232332323",M,1):IF J>B
    LET M=M+1:J=J-B:IF M>12LET M=1:A=A+1
430:GOSUB 510:IF I<>NLET J=J+N-I:GOTO 420
440:RETURN
510:I=0:IF M>2LET I=INT (30.6*M+.6)-33:GOSUB 600:I=I+F
520:IF M=2LET I=31
530:B=A-1:I=I+J+B*365+INT (B/4)-INT (B/100)+INT (B/400):
    RETURN
600:F=0:IF INT (A/4)=A/4LET F=1:IF INT (A/100)=A/100LET F=
    0:IF INT (A/400)=A/400LET F=1
610:RETURN
```

```
 4:O=(INT (30.608*(M-4+12*(M<5)))-367*(M<5)+92-(M>2)-(M>2
   AND A/4<>INT (A/4)))*(M<>1)
 5:O=O+J+N:M=1:X=0
 6:I=365+(A/4=INT (A/4)):IF O>=ILET O=O-I:A=A+1:GOTO 6
 7:I=365+(A/4=INT (A/4)):IF ABS (O-365-(A/4=INT (A/4)))>=
   ILET O=O+I:A=A-1:GOTO 7
 8:FOR I=1TO INT (O/28):X=X+VAL MID$ ("303232332323",I,1)
   +28+(A/2=INT (A/4)*I):M=M+(O>=X):NEXT I
10:J=(INT (30.608*(M-4+12*(M<5)))-367*(M<5)+92-(M>2)-(M>2
   AND A/4<>INT (A/4)))*(M<>1)
11:J=O-J:RETURN
```

FAITES DES ÉCONOMIES

Le « truc » que je vous propose, et que j'utilise déjà depuis plusieurs mois avec succès, permettra à plus d'un possesseur du PC-1500 + CE-150 de faire des économies substantielles : je n'ai pas changé mes stylos depuis le mois de MAI (NDLR : La lettre de M. ARISTIDI est datée du 13/02/86) et pourtant j'utilise intensivement mon appareil... Il suffit de remplir périodiquement les stylos avec de l'encre fraîche.

VOICI COMMENT PROCÉDER :

La structure d'un stylo est donnée en figure 1.

Cette structure a été constatée expérimentalement. L'encre à utiliser pour remplir le stylo est de la vulgaire encre de stylo à plume, en cartouche... On prendra bien sûr de l'encre bleue pour le stylo bleu, et de même pour les autres couleurs !

POUR OUVRIR LE STYLO, IL FAUT Y ALLER DÉLICATEMENT :

Se munir de deux pinces, avec la première serrer la tête du stylo, elle est assez longue pour offrir une bonne prise, et avec l'autre pince, serrer DÉLICATEMENT mais fermement, de façon à ne pas trop le déformer, le corps du stylo. Tirez fort ! !

Au bout d'un certain temps le stylo cède, et la tête se sépare du reste. On voit alors le bouchon avec un trou au milieu. Quelquefois le bouchon s'en va avec la tête. On prend alors la cartouche d'encre et on en approche l'extrémité de l'embouchure du stylo. On presse fort sur la cartouche pour en faire sortir une première goutte que l'on déposera sur l'ouverture. Avec une aiguille, on « touillera » le corps spongieux du stylo pour qu'il s'imbibe bien d'encre.

On fera aussi boire l'extrémité de la tête du stylo qui est en contact avec le corps spongieux (c'est à dire « l'autre côté » ou le « dessous » de la tête du stylo). Si une goutte ne suffit pas, on en met deux, puis on retourne le stylo. Il faut bien entendu que la tête ne dépasse pas trop du corps. On pourra comparer avec un autre stylo.

On peut ensuite essayer d'écrire avec le stylo (même à la main, ça marche) et normalement les résultats sont probants.

Et même pour les stylos très secs, cette méthode donne d'excellents résultats.

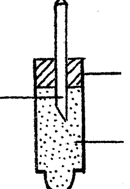
Evidemment il y a le risque d'abimer le stylo avec l'ouverture. Notons quand même qu'une boîte de 4 stylos coûte 40 F à la FNAC... et une boîte de cartouche fait environ 2,50 F, et on en met 2 gouttes à chaque fois !

Éric ARISTIDI

Dans son texte initial. M. ARISTIDI conseillait de se servir de ses incisives à la place de la deuxième pince pour éviter d'abimer le réservoir. Nous avons supprimé ce conseil estimant qu'il valait mieux perdre un stylo plutôt qu'une dent. De plus cette méthode ne marche pas avec les nouveaux stylos feutres à pointe plastique. Pour ces derniers, il est conseillé dans un bulletin de l'ACUOP de percer le stylo sur le côté pour le remplir grâce à une seringue.

LE CLUB

Figure 1



RÉCURSIVITÉ
COMMENT OPTIMALISER ?

Si vous avez lu l'article du bulletin numéro 18, vous l'aurez constaté... La récursivité ne demande pas beaucoup de calculs mais perd son temps à effectuer le passage des paramètres.
Pour avoir des programmes rapides, il faut minimiser le nombre de paramètres... Illustrons cette philosophie en réalisant ensemble le sous-programme suggéré dans le bulletin précédent (calcul du déterminant d'une matrice N x N).

Le 'truc' qui permet de calculer le déterminant d'une matrice quelconque, c'est la règle des cofacteurs. « Le déterminant d'une matrice A est égal à la somme des produits de chaque éléments d'une ligne (ou d'une colonne) par son cofacteur correspondant ». Le cofacteur de a i j (élément de A) est le déterminant de A' obtenue en supprimant de A la ligne i et la colonne j, multiplié par (−1) (i+j).

PAR EXEMPLE,

$$\text{si } A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ -9 & 10 & 11 & 12 \\ 13 & -14 & 15 & 16 \end{bmatrix}$$

et si on choisi la première colonne, on a

$$\text{dét } A = 1 * \text{dét} \begin{bmatrix} 6 & 7 & 8 \\ 10 & 11 & 12 \\ -14 & 15 & 16 \end{bmatrix} -5 * \text{dét} \begin{bmatrix} 2 & 3 & 4 \\ 10 & 11 & 12 \\ -14 & 15 & 16 \end{bmatrix}$$

$$-9 * \text{dét} \begin{bmatrix} 2 & 3 & 4 \\ 6 & 7 & 8 \\ -14 & 15 & 16 \end{bmatrix} -13 * \text{dét} \begin{bmatrix} 2 & 3 & 4 \\ 6 & 7 & 8 \\ 10 & 11 & 12 \end{bmatrix}$$

évidemment pour calculer dét $\begin{bmatrix} 6 & 7 & 8 \\ 10 & 11 & 12 \\ -14 & 15 & 16 \end{bmatrix}$ le sous-programme s'appelera lui-même et à son tour calculera,

$$6 * \text{dét} \begin{bmatrix} 11 & 12 \\ 15 & 16 \end{bmatrix} -10 * \text{dét} \begin{bmatrix} 7 & 8 \\ 15 & 16 \end{bmatrix} -14 * \text{dét} \begin{bmatrix} 7 & 8 \\ 11 & 12 \end{bmatrix}$$

etc, etc jusqu'à ce qu'il ne reste que des matrices 1 x 1 dont le déterminant est immédiat à calculer (cas de base).

Une première idée est de passer comme paramètres au sous-programme qui calcule le déterminant D, une matrice A (N,N) et sa dimension Q. Cependant, cette méthode a deux inconvénients majeurs...

1. Cela prend beaucoup de temps : à chaque appel réentrant du sous-programme, on doit réécrire une matrice entière.

2. Cela prend beaucoup de mémoire : les paramètres demandent $N + N^3$ éléments mémoires...

- 1 pour la dimension de la matrice,
- N^2 pour la matrice elle-même,

le tout N fois : n'oublions pas que chaque paramètre est un tableau !

Une autre idée est de passer comme paramètres L, les numéros des lignes et C, les numéros des colonnes sur lesquelles on doit calculer le déterminant : au lieu de travailler sur A (I,J) on travaillerait sur A (L(I), C(J)), on utilise un adressage indirect.

Il ne faut plus que $N + 2 \times N^2$ éléments en mémoire,

- 1 pour Q, la dimension du déterminant,
- N pour C, le pointeur des colonnes,
- N pour L, le pointeur des lignes...

le tout N fois, et on ne touche plus à la matrice !

Si on choisit de travailler toujours sur la dernière colonne, il n'est plus nécessaire d'indiquer quelles sont les colonnes sur lesquelles on travaille vu que se seront les Q premières... et on gagne un paramètre !

On avait déjà remarqué que la variable qui sert à indicer nos paramètres (R) pouvait faire double emploi. C'est le cas ici, vu que Q est toujours égal à N − R + 1, ... et Q n'est plus nécessaire non plus !

Enfin, la règle des cofacteurs nous suggère de ne pas calculer le cofacteur quand on devra multiplier celui-ci par zéro (ligne 110).

VOICI LE PROGRAMME.

```
10   IMPUT « DIMENSION » ; N : DIM  A (N,N), L (N, N), D (N), I (N)
20   R=0 : WAIT 0 : FOR I=1 TO N :  FOR  J=1  TO  N :  PRINT
     « A »+STR$ I+», «+STR$ J ; :  INPUT « = » ; A(I,J)
30   CURSOR :  NEXT  J :  L  (R+1, I)=I :  NEXT  I :  GOSUB  100 :
     WAIT : PRINT « DET » ; D (R) :  END
100  R=R+1 :  IF  R=N  LET  D  (R−1)=A (L(R,1), 1) : GOTO 150
110  D (R−1)=0 :  FOR I(R)=1  TO N−R+1 :  IF  A  (L(R,  I  (R)),
     N−R+1)=O GOTO 140
120  FOR I=1 TO N−R : L (R+1, I)=L  (R,I+(I(R)< =I)) : NEXT I
130  GOSUB  100 :  D  (R−1)=D (R−1)−(−1)  (I(R)+N−R)  * D
     (R) *A (L(R, I (R)), N−R+1)
140  NEXT I (R)
150  R=R−1 : RETURN
```

Ceci dit, ce programme même s'il est plus court, demande beaucoup plus de mémoire qu'un programme classique qui utilise des méthodes d'analyse numérique. Et il n'est plus rapide que pour des dimensions faibles dans certains cas particuliers...

Un cas particulier est celui des matrices qui contiennent de nombreux zéros (matrices d'incidence, etc.). Nous l'emportons sur les algorithmes de Gauss et autres parce qu'ils demandent des tests et des permutations pour éviter les divisions par zéro. Par exemple, le calcul de

$$\text{dét} \begin{bmatrix} 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10 \\ 0 & -9 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 8 & 0 & 0 \end{bmatrix}$$

ne nous demande que 4,5' pour répondre −5760... comparez avec votre programme classique, vous serez heureusement surpris !

Bon amusement, et à vos machines...

Frédéric BLONDIAU

3D HISTO

Ce programme est le premier d'une série permettant d'exploiter au mieux votre table traçante, pour des représentations graphiques classiques ou originales, à partir de tableaux de données numériques. Il est extrait d'un livre de programmes japonais et traduit pour votre plaisir.

Il est destiné aux PC 1350, 1450 et 2500 munis d'une carte 8 Ko au moins, et bien sûr d'une imprimante graphique CE-516P ou CE-140P. L'imprimante intégrée du PC 2500 fonctionne directement en supprimant l'ordre OPEN de la ligne 790. Les données sont donc représentées sous forme de graphique à barres verticales, mais en dimension 3 ce qui est moins courant (cf fig. 1). Cela donne un aspect professionnel et esthétique indiscutable, tout en autorisant une excellente lecture et interprétation. Le programme est organisé en trois parties : saisie, sauvegarde et tracé.

LA SAISIE :

Elle s'exécute par DEF A ou RUN. Un certain nombre de questions claires est alors posé. Titre (sur 10 car. max), nombre de colonnes (axe X), nombre de lignes (axe Y). L'option « Lecture Cassette » est alors proposée. En répondant O (ou o) le programme chargera les données à partir du support magnétique indiqué. Puis rendra la main en attendant de passer au tracé par DEF Z.

Autrement, on peut choisir le numéro de la couleur (0 à 3) de l'échelle (3:rouge si ENTER), et la couleur du tracé (1:bleu). Ensuite, on vous demandera le nom affecté à chaque colonne, puis à chaque ligne. On peut à présent s'occuper de la saisie numérique proprement dite (axe Z). Entrez la valeur, après l'affichage du nom de la colonne et de la ligne au signe d'invite "= >". La valeur doit être comprise entre 1 et 999999. La valeur maximale est affichée à la fin de la saisie, tapez ENTER si vous êtes d'accord ou corrigez la en fonction de l'aspect recherché, ou suivant l'échelle. Opérez de même pour la valeur minimale. En dernier lieu, vous choisissez le nombre de graduations pour l'échelle. Nous pouvons alors passer à la seconde ou troisième étape.

LA SAUVEGARDE :

L'entrée se fait exclusivement par DEF S. Si aucune saisie n'a été effectuée un message d'erreur s'affiche.

Lorsque la cassette est positionnée dans le magnétophone (position REC PLAY), tapez sur ENTER pour lancer la sauvegarde sous le nom "3DDATA".

LE TRACÉ :

C'est la partie la plus importante et celle qui demande le moins d'efforts. Comme pour la sauvegarde, si pas de saisie alors message "* PAS DE DATA *".

Positionnez auparavant le OPEN en fonction de l'imprimante, et les switches sur la CE-516P en fonction du PC. On prend l'OPEN standard pour la CE-140P (voir mode d'emploi) et pour la 516P, on peut prendre OPEN "1200,N,8,1,A,L,&1A". Le message « JE TRACE ! » indique que le tracé va s'effectuer. Si un problème survient, vérifiez que la 140P est sur PRINTER et non SIO, et que l'alimentation est suffisante. Le programme s'arrête de lui-même après le tracé du graphique et du tableau des valeurs.

Vous allez pouvoir épater les obsédés des chiffres et convaincre les plus réticents à vos études diverses.

Jean-François VIGNAUD

```
10:REM *** 3D HISTO ***
20:"A" CLEAR : WAIT 64: DIM T$(1)*20
30:INPUT "Titre=>";T$(0): GOTO 50
40:BEEP 2: GOTO 30
50:IF LEN T$(0)>10 LET T$(0)= MID$ (T$
   (0),1,10)
60:T$(1)=T$(0)
70:A=0: INPUT "Xcol(2-12)=";A: GOTO 90
80:BEEP 2: GOTO 70
90:IF A<2 OR A>12 BEEP 2: GOTO 70
100:B=0: INPUT "YLign(1-6)=";B: GOTO 12
   0
110:BEEP 2: GOTO 100
120:IF B<1 OR B>6 BEEP 2: GOTO 100
130:PAUSE "Lecture Cassette"
140:P$="": INPUT "Lecture(O/N)?";P$
150:IF P$="O" OR P$="o" GOTO 630
160:IF P$="N" OR P$="n" GOTO 190
170:BEEP 2: GOTO 140
180:REM ** COULEURS **
190:C=3: INPUT "Couleur Echel=";C: GOTO
   210
200:IF C<0 OR C>3 BEEP 2: GOTO 190
210:D=1: INPUT "Couleur Trace=";D: GOTO
   230
220:IF D<0 OR D>3 BEEP 2: GOTO 210
230:DIM X$(A-1),Y$(B-1),Z(A-1,B-1),X(A,
   B-1),Y(A,B-1)
240:FOR I=0 TO A-1
250:PAUSE "Col nom X("; STR$ (I+1);")"
260:T$(0)="": INPUT "=>";T$(0)
270:IF T$(0)="" BEEP 2: GOTO 260
280:IF LEN T$(0)>10 LET T$(0)= MID$ (T$
   (0),1,10)
290:X$(I)=T$(0)
300:NEXT I
310:FOR I=0 TO B-1
320:PAUSE "Lign nom Y("; STR$ (I+1);")"
330:T$(0)="": INPUT "=>";T$(0)
340:IF T$(0)="" BEEP 2: GOTO 330
350:IF LEN T$(0)>8 LET T$(0)= MID$ (T$(
   0),1,8)
360:Y$(I)=T$(0)
370:NEXT I
380:L=-999999:M=999999
390:FOR I=0 TO A-1: FOR J=0 TO B-1
400:PAUSE X$(I): PAUSE Y$(J)
410:INPUT "=>";Z(I,J): GOTO 430
420:BEEP 2: GOTO 410
430:Z(I,J)= INT Z(I,J): IF Z(I,J)>99999
   9 OR Z(I,J)<1 BEEP 2: GOTO 410
440:IF Z(I,J)<M LET M=Z(I,J)
450:IF Z(I,J)>L LET L=Z(I,J)
460:NEXT J: NEXT I
470:PAUSE "Val.Max="; STR$ L
480:Q$="": INPUT "Max=";Q$
490:IF Q$="" LET N=L: PRINT STR$ N:
   GOTO 520
500:IF LEN Q$>6 OR VAL Q$<L BEEP 2:
   GOTO 480
510:N= VAL Q$
520:PAUSE "Val.Min="; STR$ M
530:Q$="": INPUT "Min=";Q$
540:IF Q$="" LET O=M: PRINT STR$ O:
   GOTO 570
550:IF LEN Q$>6 OR VAL Q$>M BEEP 2:
   GOTO 530
560:O= VAL Q$
570:IF N=O LET O=0
580:INPUT "Echelle(1-10)=";E: GOTO 600
590:BEEP 2: GOTO 580
600:IF E<1 OR E>10 BEEP 2: GOTO 580
610:END
620:REM ** LECTURE K7 **
630:WAIT : PRINT "K7 OK,PRESS ENT"
640:WAIT 64: PRINT "* Lecture Data *"
650:INPUT #"3DDATA";A,B,C,D,L,M,N,O,E
660:DIM X$(A-1),Y$(B-1),Z(A-1,B-1),X(A,
   B-1),Y(A,B-1)
670:INPUT #X$(*),Y$(*),Z(*)
680:END
690:REM ** SAVE K7 **
700:"S" IF L=0 GOTO 760
710:WAIT : PRINT "K7 OK,PRESS ENT"
720:WAIT 64: PRINT "* Sauve Data *"
730:PRINT #"3DDATA";A,B,C,D,L,M,N,O,E
740:PRINT #X$(*),Y$(*),Z(*)
750:END
760:BEEP 3: WAIT 64: PRINT "* PAS DE DA
   TA *": END
770:REM ** TRACE **
780:"Z" IF L=0 GOTO 760
790:CLOSE : OPEN : CONSOLE 39: LPRINT
800:LPRINT CHR$ 27+"a": LPRINT CHR$ 13
810:LPRINT CHR$ 27+"b": LPRINT CHR$ 27+
   "0"
820:WAIT 64: PRINT "JE TRACE !"
830:LPRINT "Q1": LPRINT "M440,-100"
840:IF B<3 LPRINT CHR$ 27+"?c": GOTO 86
   0
850:LPRINT CHR$ 27+"?f"
860:LPRINT "P";T$(1): LPRINT CHR$ 27+"?
   b"
870:LPRINT "M10,-45": LPRINT "I"
880:S=N-O:V=S/E:R=210/S:F=210/E:G=210:H
   =125:W=12:YA=0
890:IF A>2 LET G=210/(A-1):H=125/(A-1)
900:FOR I=0 TO B-1: LPRINT CHR$ 27+"0"
910:LPRINT "M0,";YA: LPRINT "J210,-125"
920:Y= LEN (Y$(I))*12+25: LPRINT "R195,
   ";Y: LPRINT "P";Y$(I)
930:LPRINT "M0,";YA: LPRINT CHR$ 27+
   STR$ D
940:X(0,I)=0:Y(0,I)=YA+W:Y(1,I)=Y(0,I)
950:FOR J=0 TO A-1:X(J+1,I)= INT ((Z(J,
   I)-O)*R+G*J): NEXT J:Y=YA:YS=YA
960:LPRINT "D";X(1,I);",";YA;",";X(1,I)
   ;",";Y+W;",0,";Y+W;",0,";YA
970:FOR J=0 TO A-2
980:YE=YA+ INT ((J+1)*-H):Y(J+2,I)=YE+W
990:LPRINT "M";X(J+1,I);",";YS
1000:LPRINT "D";X(J+2,I);",";YE;",";X(
    J+2,I);",";YE+W
1010:LPRINT "D";X(J+1,I);",";YS+W
1020:YS=YE: NEXT J:S=1
1030:FOR J=A-1 TO 1 STEP -1:Y=YA+ INT
    (J*-H):X= INT (J*G)
1040:IF S=1 LPRINT "M";X(J+1,I);",";Y:
    LPRINT "D";X;",";Y: GOTO 1060
1050:LPRINT "M";X;",";Y: LPRINT "D";X(
    J+1,I);",";Y
1060:S=-S: NEXT J
1070:YA=YA-150
1080:NEXT I
1090:YA=-(B-1)*150-70: LPRINT CHR$ 27+
    "0"
1100:FOR I=0 TO A-1:X= INT (I*G)-6:Y=
    INT (YA-(I*H)-10)
```

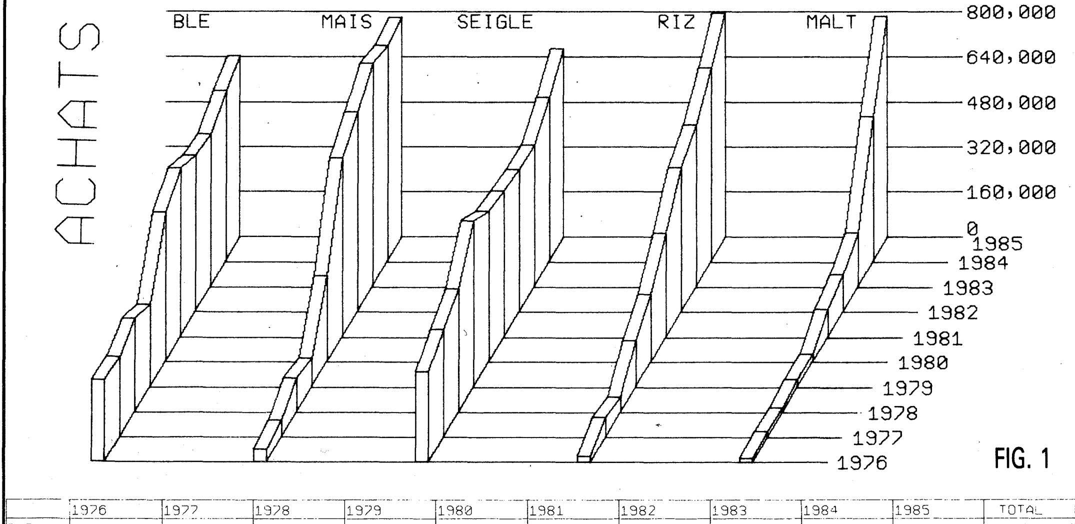


FIG. 1

	1976	1977	1978	1979	1980	1981	1982	1983	1984	1985	TOTAL
BLE	293,000	287,000	335,000	296,000	534,000	602,000	557,000	545,000	612,000	644,000	4,705,000
MAIS	47,500	62,000	120,000	105,000	310,000	640,000	716,000	797,000	772,000	782,000	4,351,500
SEIGLE	320,000	385,000	443,000	592,000	536,000	523,000	510,000	507,000	588,000	672,000	5,076,000
RIZ	23,500	22,000	55,000	170,000	245,000	373,000	516,000	578,000	692,000	799,000	3,523,500
MALT	18,500	23,000	19,000	30,000	32,000	103,000	135,000	196,000	523,000	785,000	1,864,500
TOTAL	702,500	829,000	972,000	1,193,000	1,657,000	2,241,000	2,434,000	2,623,000	3,187,000	3,682,000	19,520,500

```
1110:IF I=A-1 AND I<>0 LET X=198
1120:LPRINT "M";X;",";Y: LPRINT "P";X$
     (I)
1130:NEXT I
1140:LPRINT CHR$ 27+ STR$ C
1150:Y=YA-130: FOR I=0 TO E:X= INT (I*
     F-6+210): IF I=0 LET X=212
1160:K= INT (O+I*V): GOSUB 1200
1170:LPRINT "M";X;",";Y: LPRINT "P";Q$
1180:NEXT I: GOTO 1220
1190:REM ** NOTAT FIN **
1200:P$= STR$ K:T= LEN P$: IF T<4 LET
     Q$=P$: RETURN
1210:Q$= LEFT$ (P$,T-3)+","+ RIGHT$ (P
     $,3): RETURN
1220:GOSUB 1590
1230:GOSUB 1800
1240:LPRINT "M15,";YA
1250:LPRINT "R0,";-300: LPRINT CHR$ 27
     +"0": LPRINT "I"
1260:REM ** TABLEAU **
1270:T=25*(B+2):U=100: LPRINT CHR$ 27+
     "?b"
1280:LPRINT "J0,";-U;",";T;",0,0,";U;"
     ,";-T;",0":S=1
1290:FOR I=1 TO B+1:X=I*25
1300:IF S=1 LPRINT "M";X;",0": LPRINT
     "D";X;",";-U: GOTO 1320
1310:LPRINT "M";X;",";-U: LPRINT "D";X
     ;",0"
1320:S=-S: NEXT I:X=T-20:Y=-3
1330:FOR I=0 TO B-1:X=X-25: LPRINT "M"
     ;X;",";Y: LPRINT "P";Y$(I): NEXT
     I
1340:LPRINT "M5,-3": LPRINT "P";"TOTAL
     ": LPRINT "M0,";-U: LPRINT "I"
1350:U=120: FOR I=0 TO A: LPRINT "D0,"
     ;-U;",";T;",";-U;",";T;",0"
1360:S=1: FOR J=B TO 0 STEP -1:X=25*(J
     +1)
1370:IF S=1 LPRINT "M";X;",0": LPRINT
     "D";X;",";-U: GOTO 1390
1380:LPRINT "M";X;",";-U: LPRINT "D";X
     ;",0"
1390:S=-S: NEXT J:X=T-20
1400:IF I=A GOTO 1520
1410:Y=-3: LPRINT "M";X;",";Y: LPRINT
     "P";X$(I)
1420:TT=0: FOR J=1 TO B:X=X-25:K=Z(I,J
     -1): GOSUB 1460: NEXT J
1430:X=3:K=TT: GOSUB 1460: LPRINT "M0,
     ";-U: LPRINT "I"
1440:NEXT I: GOTO 1530
1450:REM ** NOTAT FIN **
1460:TT=TT+K:ST$= STR$ K:W1= LEN ST$:
     IF W1<4 GOTO 1500
1470:ST$= LEFT$ (ST$,W1-3)+","+ RIGHT$
     (ST$,3)
1480:W1= LEN ST$: IF W1<8 GOTO 1500
1490:ST$= LEFT$ (ST$,W1-7)+","+ RIGHT$
     (ST$,7)
1500:W1= LEN ST$:Y=(10-W1)*-12-2
1510:LPRINT "M";X;",";Y: LPRINT "P";ST
     $: RETURN
1520:REM ** TOTAL **
1530:LPRINT "M";X;",";-20: LPRINT "P";
     "TOTAL":TT=0
1540:FOR J=1 TO B:X=X-25:K=0: FOR I=0
     TO A-1:K=K+Z(I,J-1): NEXT I
1550:GOSUB 1460: NEXT J
1560:X=3:K=TT: GOSUB 1460
1570:LPRINT "M-25,";-U-100: LPRINT
     CHR$ 27+"a"
1580:CONSOLE 39: LPRINT : CLOSE : END
1590:YS=0: LPRINT CHR$ 27+"0"
1600:IF B=1 THEN 1740
1610:FOR J=1 TO B-1: LPRINT "M0,";YS
1620:IF B>1 LPRINT "J0,-150"
1630:IF A=1 AND B>1 LPRINT "M210,";YS-
     125: LPRINT "J0,-150": GOTO 1720
1640:IF A=1 AND B=1 THEN 1720
1650:FOR I=1 TO A-1:X= INT (I*G):YE=YS
     - INT (I*H)
1660:FOR K=0 TO A-1
1670:IF X(K,J)>X OR X(K+1,J)<=X THEN 1
     700
1680:S=(Y(K+1,J)-Y(K,J))/(X(K+1,J)-X(K
     ,J)):Y= INT (S*X-S*X(K,J)+Y(K,J))
1690:LPRINT "M";X;",";YE: LPRINT "D";X
     ;",";Y: GOTO 1710
1700:NEXT K: LPRINT "M";X;",";YE: IF B
     >1 LPRINT "J0,-150"
1710:NEXT I
1720:YS=YS-150: NEXT J:S=1
1730:IF B=1 LET YS=YS+150
1740:FOR I=0 TO A-1:X= INT (I*G):Y=YS-
     INT (I*H)
1750:IF S=1 LPRINT "M";X;",";Y: LPRINT
     "D";X;",";Y-70: GOTO 1770
1760:LPRINT "M";X;",";Y-70: LPRINT "D"
     ;X;",";Y
1770:S=-S: NEXT I
1780:IF A=1 LPRINT "M210,";YS-125:
     LPRINT "J0,-70"
1790:RETURN
1800:YS=-55:YE=YS-(B-1)*150-140:
     LPRINT CHR$ 27+ STR$ C
1810:FOR K=1 TO E:X=210+ INT (K*F):Y1=
     YS
1820:FOR J=0 TO B-1: FOR I=0 TO A-1
1830:IF X(I,J)>X OR X(I+1,J)<=X THEN 1
     880
1840:S=(Y(I+1,J)-Y(I,J))/(X(I+1,J)-X(I
     ,J)):Y= INT (S*X-S*X(I,J)+Y(I,J))
1850:IF Y>Y1 THEN 1870
1860:LPRINT "M";X;",";Y1: LPRINT "D";X
     ;",";Y
1870:Y1=-(J*150)-125
1880:NEXT I: NEXT J
1890:LPRINT "M";X;",";Y1: LPRINT "D";X
     ;",";YE
1900:NEXT K: RETURN
1910:CONSOLE 39: CLOSE : END
```

4638 octets

ON

Grâce à l'article du n° 19 vous avez appris à mettre en marche votre PC, maintenant nous allons utiliser les variables (de plus en plus fort le Sharpentier !). Avant de se lancer dans des considérations techniques en distinguant les différents types de machines, nous allons d'abord étudier ce qu'elles ont de commun ; voyons d'abord comment s'utilisent les variables en mode calcul. On distingue deux sortes de variables. Les numériques ne peuvent contenir que des chiffres, et les alphanumériques peuvent stocker toutes les sortes de caractères que l'ordinateur connaît.

NUMÉRIQUES

On se sert généralement de ces variables pour stocker les résultats intermédiaires dans les calculs, c'est dire leur importance !

Nous allons prendre un exemple général avec une variable de nom A. En effet les variables ont des noms qui correspondent aux lettres de l'alphabet. Vous en avez donc 26 directement accessibles, mais nous verrons plus tard qu'en fait vous en avez bien plus.

Il y a deux actions courantes que l'on fait subir à une variable :
- lui affecter une valeur, ou
- lire sa valeur et l'utiliser dans un calcul.

Le premier cas est assez simple, il suffit d'écrire le nom de la variable, de mettre un signe égal, et d'écrire la valeur ou le calcul dont la valeur s'affectera à cette variable.

Par exemple :
A = 10 ENTER
Revient au même que
A = 5*2 ENTER

Maintenant, lorsque vous voudrez lire cette valeur, il vous suffira de taper :
A ENTER
et 10 s'affichera.

Mais vous pouvez aussi utiliser la valeur 10 contenue dans A à l'intérieur d'un calcul. Ainsi :
A*5 ENTER
donnera 50, ce qui correspond bien à A (= 10)*5. Mais vous pouvez stocker ce résultat dans une autre variable, par exemple B. On fera alors :
B = A*10 ENTER
et B vaudra bien 50.

On peut aussi affecter à une variable une valeur qui est calculée à partir de sa valeur précédente.

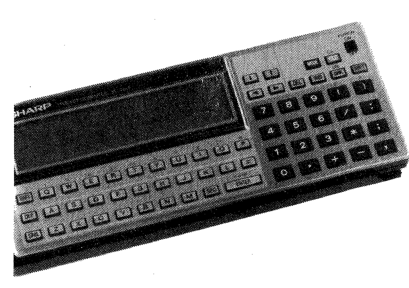
Par exemple :
A = A + 1 ENTER
effectuera le calcul A + 1, dans notre cas A valant toujours 10, ce calcul vaudra 11 et le résultat sera de nouveau stocké dans la variable de nom A qui perdra ainsi son ancienne valeur.

ALPHANUMÉRIQUES

Ces variables se comportent différemment, en effet, on ne traite plus des chiffres mais des caractères. Ainsi on pourra mettre en mémoire des mots. Les principales règles d'écriture pour ces variables sont les suivantes :
- un signe $ après le nom de la variable indique qu'elle sera alphanumérique
- les mots ou message sont entre guillemets
- seule l'addition est possible.

Par exemple :
A$ = "NOM" ENTER - stocke le message NOM dans la variable A$
A$ ENTER - vous lisez la variable de nom A$
A$ + "INE" ENTER - vous rajoutez au message NOM la partie INE et le résultat est logiquement NOMINE.
Si vous aviez fait :
B$ = "NOM" + A$ ENTER
vous auriez dans B$ le message NOMNOM, et pour vous en convaincre : B$ ENTER.
Vous pouvez deviner qu'avec cette syntaxe, il est impossible de mettre un guillemet dans un message. Gênant ! Remédions-y grâce à une merveilleuse instruction : CHR$.
En règle générale, lorsqu'une instruction possède un $ comme dernier caractère, sa réponse sera une chaîne de caractères. Dans notre cas précis CHR$ est l'abréviation de CHaracterR$. Tous les caractères que vous pouvez afficher sur votre écran sont en fait codés en mémoire par un numéro selon un code appelé code ASCII qui se traduit en gros par « Code Standard et International » !
Vous ouvrez vos grands yeux ! Du standard en informatique ! Impossible me direz-vous ! et je vous réponds bien sûr, seuls les codes les plus courants ont presque partout le même numéro. Ainsi A est 65, et " est 34. Ainsi, si vous faites CHR$ 34 ENTER, un superbe " vient s'offrir à vous. Il ne me reste qu'à l'intégrer à un message.
A$ = "NOM" + CHR$34 + "PC"
+ CHR$34 ENTER
et le résultat sera :
NOM "PC". Superbe non ? !
Et toujours plus fort...
Et bien non ! Le rédacteur en chef m'informe qu'il est temps de conclure sans quoi je déborderais de mon espace dévolu. La prochaine fois nous attaquerons donc MID$, LEN, STR$ et toute une série d'autres choses amusantes.

Le Club



OLD POUR PC 1350, 2500, 1260/61, 1401/02/50

Dans le n° 17, je vous avais proposé RENEW pour récupérer un programme BASIC « effacé » par un NEW ou un ALL RESET. Ce programme LM n'exploitait pas la structure des lignes Basic propres aux PC cités. Par le fait, OLD donne le même résultat plus rapidement, et occupe 23 octets seulement.

Les Dumps fournis pour chaque série se rentrent aisément par POKE adresse, data, data... Ex. pour le PC 1350, on tapera POKE &6E48,&F4,&71,&04,&02,&00,&52,&04,&24. Puis POKE &6E50,&03,&00,&84 etc. Pour le Dump du 1350, le nombre hexa après les deux points est un contrôle de cheksum, utilisable par l'utilitaire donné pour cette machine dans notre n° 13. Pour les PC autres que 14XX, le programme est situé à la fin de la zone système des boucles : tampon des FOR-NEXT. Les PC 1401/02/50 ont la désagréable habitude (ce qui m'avait échappé pour RENEW, mais pas aux utilisateurs de PC 14XX, comme le témoigne vos lettres) de réinitialiser ses tampons à l'allumage : Sûrement à cause du mode CAL. J'ai donc placé le programme LM ailleurs (à la fin de la zone Basic). De toutes façons, le LM est relogeable et peut donc être POKé où l'on veut sans modifications. ATTENTION ! les possesseurs de 1401 doivent faire un POKE &46FC,&B8,&45 (POKE &5F07,&18,&5C pour le 1450), pour protéger le LM des DIM qui l'écraserait à coup sûr. Après un NEW, ALL RESET ou CLEAR, refaire ce POKE. Le mieux est de rechercher un emplacement de 23 octets fiable et sûr. Le désassemblage fourni doit normalement vous permettre de comprendre le fonctionnement de ce programme court, mais fort utile. En rappel voici la structure d'un programme BASIC dans ces machines :

PAR EXEMPLE :
Ce programme dans un PC 1350 sans carte.

```
10 PRINT A
20 END
```

Le programme ci-dessus est codé :

	Address	Data	
&6030→ START BASIC	6030H	FFH	... code début prog BASIC
	31H	00H	10 n° de ligne (PF, Pf)
	32H	0AH	
&6033 + 3 = &6036	33H	03H	... Longueur de la ligne
	34H	DEH	... Token PRINT
	35H	41H	... A
	36H	0DH	... CR
	37H	00H	20
	38H	14H	
	39H	02H	... Longueur ligne
	3AH	D8H	... END
	3BH	0DH	... CR
&603C → END BASIC	603CH	FFH	...Code fin prog BASIC

Les deux FF encadrant le programme sont à des adresses conservées dans les pointeurs appelés : START BASIC et END BASIC. Lors d'un NEW, le second FF vient se placer à la suite du premier (détruisant le premier octet du numéro de ligne) et END BASIC est initialisé avec cette nouvelle adresse. Le programme LM place 0 sur ce FF et recherche l'adresse de l'ancien FF pour remettre dans END BASIC, l'adresse qu'il contenait avant le NEW. Il va sans dire que le programme Basic n'est pas réellement effacé.

Jean-François VIGNAUD.

```
1000*******
1010* OLD *
1020*******
1030/ORG &6E48
1040*
1050        CAL   &1471
1060        IX
1070        LIA   &00
1080        STD
1090(LBL1)  IX
1100        IXL
1110        LIB   &00
1120        LP    &04
1130        ADB
1140        IXL
1150        CPIA  &FF
1160        JRNZM (LBL1)
1170        LIDP  &6F03
1180        LP    &04
1190        EXBD
1200        RTN
1210/ END
```

START BASIC → X
On se place sur le 1er octet du numéro de la ligne (PF).
On met la valeur 0 à la place du second FF dû au NEW.
On se place sur le 2e octet du numéro.
On met dans A la longueur de la ligne.
0 → B ⟹ 00LEN → BA
BA + X → X on se place sur le 0D de fin de ligne.
On regarde si le code suivant est l'ancien FF de fin de programme.
Si non on recommence car c'est un numéro de ligne.
Si oui ont réinitialise END BASIC avec l'adresse trouvée dans X.
Et c'est tout !

```
DUMP  OLD  PC 1350
6E48 F471040200520424:E5
6E50 030084142467FF29:4E
6E58 0A106F03841B3700:62

DUMP  OLD  PC 1260/61
6748 F3BD 0402 0052 0424
6750 0300 8414 2467 FF29
6758 0A10 66E3 841B 3700

DUMP  OLD  PC 2500
6F48 F555 0402 0052 0424
6F50 0300 8414 2467 FF29
6F58 0A10 6D93 841B 3700

DUMP  OLD  PC 1401/02
45B8 F643 0402 0052 0424
45C0 0300 8414 2467 FF29
45C8 0A10 46E3 841B 3700
```

```
DUMP  OLD  PC 1450
5C18 FA31 0402 0052 0424
5C20 0300 8414 2467 FF29
5C28 0A10 5F03 841B 3700
```



DES ERREURS... QUI N'EN FAIT PAS ?

NUMERO 12 • PAGE 36 • **SUBMARINER**

La ligne 1010 est :
DATA 95,67,67,103,113,97,71,115,67,97,115,71,71,99,65

NUMERO 14 • PAGE 30 • **NAVIGATION ASTRO**

Ligne 5 ; 51 ; 62 ; 304 ; 306 ; 752 ; remplacer SQR par √
Ligne 304 et 498 ; remplacer PI par π
Ligne 59 ; mettre en dernier caractère Y

NUMERO 15 • PAGE 21 • **PC BLUES** • COL. 2

Pour charger PC BLUES :
NEW PEEK &7864 * 256 — 2200

NUMERO 17 • PAGE 14 • **SURFACES**

Suite et fin du programme « SURFACES »

```
1510: S=2×π×R×H
1515: GOTO 115
1600:"16":CLEAR:PRINT "SURFACE D UN..":PRINT "FU
     SEAU...":PRINT "TELLE QUE ...":PRINT "S=(π×R^2×
     N)/90"
1605:INPUT"R:  ";R:INPUT"N:  ";N
1610: S=(π×R^2×N)/90
1615: GOTO 115
1650: END
```

NUMERO 17 • PAGE 54 • **HARD COPY** • COL.3

Ligne 1010 :
O = POINT (A,B) : C = E*B, D = E*A : IF O = 0 NEXT B : NEXT A : END

NUMERO 18 • PAGE 28 • **CLASSEMENT** • COL. 2

ayant au maximum deux caractères, le second ne pouvant être un chiffre.
SGN OR = O ou = 1 tri croissant

Listing hexadécimal
adresse 18A:&50 et non &4F

routine de relogement :

```
10 DATA &D1,&DD,&E3,&E6,&FC,&FF,&103,&150,&166,&16A
20 DATA &174,&190,&193,&196,&19A,&1A6,&1AF,&1B5,&1BC,&1E9
30 DATA &200,&20D,&215,&219,&22B,&265,&269,&26D,&270,&275,&278,&27E
40 INPUT "Adresse(nnC5):";A:IF (AAND &FF)<>&C5THEN 40
50 B=INT (A/256):RESTORE
60 FOR I=1TO 32:READ C:POKE B*256+C,PEEK (B*256+C)+B:NEXT I
70 BEEP 5
```

NUMERO 18 • PAGE 29 • **INITIATION LH-5801** • COL. 3

```
10 WAIT 0:FOR I=7TO 0STEP -1:PRINT STR$ (A>=2^I);:A=A-(2^I)*(A>=
   2^I):NEXT I:WAIT :PRINT
```

NUMERO 18 • PAGE 30 • COL. 3

```
10 A=0:FOR I=1TO 8:A=A+2^(8-I)*VAL MID$ (A$,I,1):NEXT I:PRINT A
```

NUMERO 18 • PAGE 31 • COL. 1

si nm <O 256-nm

NUMERO 18 • PAGE 36 • **NAVIGATION ASTRO** • COL. 3

46 : G = H/24 — P + Z/B + (Q—A + X)/R — 0.000005*SIN (R*(T—5)/29.53))

NUMERO 19 • PAGE 8 • **DUMP REGISTRES**

Rien qu'un petit listing oublié

```
10:REM DUMP RAM INTERNE
   ,FLAGS,REGISTRES
20:POKE &6900,&10,&69,&
   3F,&52,&11,&40,&02,&
   01,&38,&03,&02,&00,&
   52,&11,&41,&02
30:POKE &6910,&01,&3A,&
   03,&02,&00,&52,&11,&
   42,&20,&52,&11,&43,&
   21,&52,&11,&44
40:POKE &6920,&22,&52,&
   11,&45,&80,&53,&00,&
   5D,&11,&46,&81,&19,&
   11,&46,&81,&18
50:POKE &6930,&11,&45,&
   80,&55,&11,&42,&57,&
   30,&11,&3F,&57,&11,&
   47,&52,&37
55:DIM B$(0)*24: CALL &
   6900
60:PRINT "Z  C  P  Q  R
   "
70:FOR I=&6940 TO &6944
   :X= PEEK I: GOSUB 20
   0: NEXT I: PRINT B$(
   0)
75:B$(0)=""
80:PRINT "I  J  A  B  X
   | Xh Y| Yh": FOR I=&
   6945 TO &694C:X=
   PEEK I: GOSUB 200:
   NEXT I: PRINT B$(0)
85:B$(0)=""
90:PRINT "K  L  M  N  0
   c 0d 0e 0f": FOR I=&
   694D TO &6954:X=
   PEEK I: GOSUB 200:
   NEXT I: PRINT B$(0)
95:PRINT ""
100:FOR I=&6955 TO &69A4
   STEP 8:B$(0)="":
   FOR J=0 TO 7:X=
   PEEK (I+J): GOSUB 20
   0: NEXT J: PRINT B$(
   0): NEXT I
110:END
200:Y= INT (X/16):Z=X-Y*
   16:B$(0)=B$(0)+
   CHR$ (Y+&30+(Y>9)*7)
   + CHR$ (Z+&30+(Z>9)*
   7)+" "
210:RETURN
```

NUMERO 19 • PAGE 3 • **ET TOUJOURS HAD**

Toutes nos excuses à ceux qui n'auraient pas le module mémoire adéquat pour faire fonctionner MAD dans la version d'origine. Comme vous avez dû le remarquer, le listinq du n° 19 a « sauté ».

Après avoir fait tous les POKEs du n° 19, mettre le LM à une adresse du type &NN00 donc ayant le poids faible nul, faire NEW &NN00 + &1200. Et taper le programme Basic de relogement, RUN, et à la question, répondez par &NN00. Quelques secondes plus tard, le programme sera relogé.

Le Club.

```
 5:DATA &A40,&6F1,&41,&5B,&7C,&AD,&BC,&76F,&9A5,&AE2,&AF3
 10:DATA 1,4,&D,&14,&1B,&20,&31,&36,&46,&4B,&56,&67,&81,&8
    6,&89,&8C,&8F
 20:DATA &94,&9A,&A8,&B0,&B3,&DC,&EA,&FD
 30:DATA &634,&642,&670,&686,&6A0,&6A6,&6AE
 40:DATA &728,&751,&757,&75C,&75F,&762,&790,&79E,&7C2,&7C9
    ,&7E5,&7EC,&7F0,&7F3
 50:DATA &809,&825,&82D,&830,&838,&878,&887,&88A,&88D,&899
    ,&8A2,&8A9,&8AC,&8B0
 60:DATA &8B8,&8BB,&8E7,&8ED,&8F3,&8F7,&8FC,&901,&907,&90A
 70:DATA &910,&913,&918,&91C,&923,&927,&92B,&930,&933,&937
    ,&94B,&958,&95B,&960
 80:DATA &96E,&97B,&97E,&985,&988,&98B,&994,&99C,&9A8,&9AB
    ,&9B3,&9B6,&9BD
 90:DATA &9C2,&9C8,&9CD,&9DB,&9E0,&9E3,&9EE,&A08,&A10,&A18
    ,&A20,&A28,&A30,&A38
100:DATA &A40,&A58,&A61,&A64,&A67,&A6E,&A71,&A74,&A7D
110:DATA &A81,&A8A,&A8D,&A92,&A95,&A98,&A9B,&A9E,&AA6,&AA9
    ,&AAC,&AAF
```

ERREURS... QUI N'EN FAIT

```
120:DATA &AB2,&AB9,&AC4,&AC7,&ADO,&ADA,&ADD,&AE5,&AE8,&AEB
130:DATA &AF6,&AF9,&BO1,&BOB,&BOE,&B13,&B16,&B22,&B2C,&B57
    ,&B5B,&B73,&B7A,&BB3
140:DATA &BB7,&BBD,&BC8,&BCC,&BD1,&BD4,&BD8,&BDB,&BEB,&BF5
    ,&BFD
150:DATA &CO1,&CO8,&COB,&COE,&C11,&C14,&C1E,&C21,&C2D,&C38
    ,&C44,&C4F,&C5A,&C5F
160:DATA &C6B,&C78,&C7E,&C83,&C86,&C89,&C8D,&C99,&CA6,&CB1
    ,&CBE
170:DATA &CC3,&CC8,&CDO,&CD3,&CD6,&CDB,&CE1,&CE6,&CEB,&CFC
    ,&DO3,&D15,&D2D,&D38
180:DATA &D53,&D56,&D5C,&D68,&D6D,&D70,&D74,&D77,&D7C,&D86
    ,&D8D,&D98,&D9D,&DAO
190:DATA &DA5,&DAF,&DB4,&DBF,&DC4,&DD8,&DDF,&DE2,&DE8,&DF4
    ,&DF9,&DFC,&E01,&E04,&E08
200:DATA &EOD,&E12,&E15,&E1F,&E2F,&E37,&E3D,&E40,&E45,&E49
    ,&E4C,&E54,&E60,&E6E
210:DATA &E7B,&E80,&E85,&E8D,&E99,&E9C
220:DATA &F20,&F28,&F31,&F35,&F3D,&F42,&F5A,&F62,&F6E,&F75
    ,&F7D,&F85,&F90,&F96,&F9E
230:DATA &FA3,&FAB,&FBO,&FB8,&FBD,&FCO,&FC8,&FD1,&FDA,&FDF
    ,&FE7,&FEE,&FF3
240:DATA &1001,&1011,&101A,&1023,&102C,&1031,&1039,&1041,&
    1048,&1064,&106C,&107F
250:DATA &1084,&108C,&1091,&1094,&10A2,&10AD,&10B4,&10BD,&
    10C2,&10CA,&10D3,&10DC
260:DATA &10E1,&10E9,&10EE,&10F1,&10F9,&10FD
270:ON ERROR GOTO 290:RESTORE :INPUT "@:";A:IF AAND &FF<>O
    THEN 270
280:READ I:I=I+A:POKE I,PEEK I-&4F+(A/256):GOTO 280
290:END
```

NUMERO 19 • PAGE 4 • **MONSTRES ET CLÉS**

Un cafouillage « MONSTRE » dans le montage du listing à partir de la ligne 120.

```
  1:P=50: REM MONSTRES&C
    LE
  2:RANDOM :T=0
 10:WAIT 80: PRINT "INIT
    IALISATION": DIM Z(6
    ,6): FOR I=0 TO 6:Z(
    I,0)=9:Z(I,6)=9:
    NEXT I
 20:FOR I=0 TO 6:Z(0,I)=
    9:Z(6,I)=9: NEXT I:
    FOR I=1 TO 5: FOR J=
    1 TO 5
 30:Z(I,J)= RND 6: NEXT
    J: NEXT I
 35:Z( RND 5, RND 5)=7
 40:X= RND 5:Y= RND 5:
    IF Z(X,Y)=7 THEN 40
 50:Z(X,Y)=8
 60:X= RND 5:Y= RND 5:
    IF (Z(X,Y)=7)+(Z(X,Y
    )=8) THEN 60
 70:PRINT "COORD.";X;"/"
    ;Y: INPUT "N/S/E/O ?
    ";A$: IF A$<>"N" AND
    A$<>"S" AND A$<>"E"
    AND A$<>"O" THEN 70
 80:IF A$="N" LET B=-1:D
    =0
 90:IF A$="S" LET B=1:D=
    0
100:IF A$="E" LET B=0:D=
    1
110:IF A$="O" LET B=0:D=
    -1
120:IF Z(X+D,Y+B)=9
    PRINT "UN MUR !!":
    GOTO 70
130:X=X+D:Y=Y+B: IF Z(X,
    Y)=7 PRINT "VOUS TRO
    UVEZ": PRINT "LA CLE
    ":T=1:Z(X,Y)=0: GOTO
    70
140:IF Z(X,Y)=8 AND T=1
    PRINT "VOUS AVEZ GAG
    NE": PRINT "BRAVO":
    END
145:IF Z(X,Y)=5 PRINT "U
    NE POTION":P=P+3:
    PRINT "VOUS AVEZ ";P
    ;"PV":Z(X,Y)=0: GOTO
    70
146:IF Z(X,Y)=0 AND RND
    7=4 GOTO 155
150:IF Z(X,Y)<2 PRINT "R
    IEN A SIGNALER":
    GOTO 70
155:PRINT "UN MONSTRE"
160:RANDOM :K= RND 5:
    PRINT "VOUS PERDEZ";
    K;"PV":P=P-K: IF P<=
    0 PRINT "VOUS ETES M
    ORT": PRINT "FIN":
    END
170:IF RND 5>2 PRINT "VO
    US LE TUEZ":Z(X,Y)=0
    : GOTO 70
175:PRINT "VOUS AVEZ ";P
    ;"PV"
180:PRINT "VOUS LE RATEZ
    ": PRINT "IL ATTAQUE
    ": GOTO 160
```


MODIFICATIONS POUR LE K-BASIC V.5.9 (voir n° 17 p. 60)

Charger le Basic, puis lorsque la sonnerie retentit, retourner au moniteur par Shift Break. Par la commande M, changer les valeurs des octets.

1277 : 04	127D : 7D	1285 : 55	4545 : C9	1OFD : 31
1278 : 6B	127E : 06	1286 : 06		1OFE : 3O
				1OFF : OD

Puis, toujours par M, entrer à partir de 7 DO4 les octets suivants :

3504 : C3	714B : DA	2E73 : C3
3505 : OD	714C : 37	2E74 : 45
3506 : 6B	714D : 6B	2E75 : 6B

21 15 00 22 5D 00 C3 50 33 CA 8B 23 FE E6 C2 07 23 CD 83 53 CD 46 48 FE 28
D2 A6 20 32 54 00 7E FE 29 20 03 23 18 06 CD 36 68 CD 8E 53 CD 42 4F C3 07
23 3A 12 73 CB 07 D2 A9 20 D1 CD E4 56 C1 C9 7E FE 21 28 05 FE 3B C3 76 1C
CD 84 1D 23 C3 6D 1C (Note : le dernier octet est entré dans la mémoire 7D55)

On réenregistrera le Basic par J8670 et enfoncement de S.

Le K-Basic V.5.10 aura ainsi les nouvelles possibilités du Super Basic 6.2 avec de plus,

CTRL H qui remet CONSOLE au maximum et réinitialise les couleurs de l'écran (il faudra cependant faire CTRL V ou CLS pour effacer l'écran).

CTRL K (ou PRINT CHR$ (11)) qui fait descendre l'écran d'une ligne.

CTRL O (ou PRINT CHR$ (15)) qui fait remonter l'écran d'une ligne.

Bernard KOKANOSKI

LES CARACTÈRES DE CONTRÔLE

A part les quelques privilégiés qui utilisent le MZ-700 sous CP/M, la majorité des Sharpentiers ne se sert pas de la touche CTRL (chez certains, la poussière commence même à s'accumuler dessus !). C'est d'autant plus dommage que cette touche, convenablement utilisée, peut faciliter beaucoup la tâche du programmeur.

Elle s'utilise comme SHIFT, c'est-à-dire que seule elle n'a aucun effet. Il faut appuyer en même temps sur une autre touche. On peut distinguer quatre effets différents suivant cette touche :

- avec ALPHA elle permet de repasser en majuscule si vous étiez passé en minuscule par SHIFT ALPHA,
- avec les touches de déplacement du curseur, INST et DEL, CTRL semble n'avoir aucun effet ; on obtient des déplacements de curseur, des insertions et des effacements, comme si elle n'était pas enfoncée. Cela s'avère pourtant très utile car, quand on est en mode GRAPH, on ne peut plus bouger le curseur. Il suffit donc d'appuyer conjointement sur CTRL et une de ces touches jaunes pour retrouver sa liberté,
- avec les lettres de l'alphabet majuscule et [, \,], ↑, / on obtient les carac-

tères de contrôle et c'est là l'utilité principale de CTRL (CTRL est d'ailleurs l'abréviation de ConTRôLe). Nous y reviendrons...

• avec toutes les autres touches on obtient un écho de la dernière touche enfoncée, cela ne sert strictement à rien et peut même être considéré comme une erreur des concepteurs du S-BASIC !!!

Mais qu'est-ce donc que ces fameux caractères de contrôle que l'on peut obtenir avec CTRL ?

Ils n'ont de caractère que le nom car ils ne correspondent pas à un symbole affichable, mais à un effet particulier sur le fonctionnement des entrées/sorties de la machine (saut de ligne, effacement de l'écran, blocage du clavier en minuscule, etc.). Ils correspondent aux codes ASCII de 0 à 31.

Leur utilité était plus manifeste avant l'avènement de la micro. En effet, l'ordinateur et la console d'entrée/sortie étaient alors deux machines distinctes reliées par un cable, l'ordinateur envoyait donc à la console soit des caractères affichables (code ASCII > = 32) ou des caractères de contrôle (ASCII < 32). Ainsi, si l'ordinateur voulait un bip, il envoyait le code « bip » (ASCII = 7) à la console qui disposait d'un haut-parleur inexistant sur l'ordinateur.

Voici donc un tableau des caractères de contrôle du MZ-700 :

ASCII	TOUCHE	ROLE
0		
1	CTRL A	}
2	CTRL B	} inutilisé
3	CTRL C	}
4	CTRL D	}
5	CTRL E	passage en minuscules
6	CTRL F	passage en majuscules
7	CTRL G	inutilisé (sauf K-BASIC)
8	CTRL H	
9	CTRL I	}
10	CTRL J	} inutilisé
11	CTRL K	}
12	CTRL L	}
13	CTRL M	idem touche CR
14	CTRL N	inutilisé
15	CTRL O	inutilisé
16	CTRL P	idem touche DEL
17	CTRL Q	idem touche ↓
18	CTRL R	idem touche ↑
19	CTRL S	idem touche →
20	CTRL T	idem touche ←
21	CTRL U	comme touche HOME
22	CTRL V	comme touche CLR
23	CTRL W	comme touche GRAPH
24	CTRL X	comme touche INST
25	CTRL Y	passage en majuscule (?)
26	CTRL Z	inutilisé
27	CTRL [	curseur début ligne suivante
28	CTRL \	}
29	CTRL]	} inutilisé
30	CTRL ↑	}
31	CTRL /	}

On constate donc que beaucoup de ces caractères sont inutilisés. Mais il est possible de leur attribuer un rôle relativement facilement.

En effet, quand le MZ-700 doit « afficher » un caractère de contrôle, il va exécuter le sous-programme LM dont l'adresse est située en $0067 + 2 * Code ASCII du caractère de contrôle. Il suffit donc de modifier la table qui s'étend de $0067 à $00A6 pour attribuer de nouvelles fonctions aux codes inutilisés.

C'est ce que fait le court programme BASIC que nous vous proposons. Exemple, si l'on associe au code 31 (CTRL /) l'adresse $209F (qui est celle du sous-programme CONT), on pourra poursuivre l'exécution d'un programme après un BREAK, par CTRL /. Pour vous éviter de fastidieux calculs d'adresse, vous pouvez utiliser le petit programme joint à cet article. Quant à lui, le listing hexadécimal représente 4 routines qui bien que minuscules s'avèrent à l'usage très utiles. De plus elles sont complètement relogeables, c'est-à-dire que vous pouvez les implanter n'importe où en mémoire (ici en $2CC0 zone inusitée au cœur du S-BASIC)

INVERSE : situé au début du listing (déplacement $00) a pour rôle d'inverser les deux pages d'écran, cela évite d'appuyer 25 fois sur SHIFT ↑.

NETTOIE : (en déplacement $28) restaure la couleur et la taille d'écran initiales. Cela évite d'avoir à recharger le BASIC quand on a trop rétréci la fenêtre d'écran pour pouvoir taper CONSOLE en entier, ou de pédaler dans la choucroute quand le COLOR choisi donne des caractères noirs sur fond noir.

CONTRASTE : (en déplacement $31) permet de passer en inverse vidéo pour mettre en valeur une portion de texte.

AGRANDI : (situé en déplacement $3E) remplace le jeu de caractère standard par le deuxième générateur, et donc permet d'obtenir des lettres agrandies ou des symboles graphiques absents du premier jeu. Pour utiliser ces routines, entrez-les sous forme hexadécimale avec le moniteur du BASIC. Puis exécutez 4 fois le petit programme BASIC pour associer les caractères de contrôle aux routines (vous pouvez utiliser CTRL I, CTRL N, CTRL C, CTRL A comme j'ai choisi de le faire ou n'importe quelle autre touche inutilisée). Ensuite, sauvegardez votre BASIC modifié si vous voulez disposer définitivement de cette amélioration.

Dès lors, ces routines sont accessibles directement au clavier et peuvent même être utilisées dans un programme en faisant des PRINT CHR$ (XX), où XX est l'un des codes choisis.

Simon CHAGNOUX

```
:2CC0=F5 E5 D5 C5 D3 E3 11 00
:2CC8=D0 01 E8 03 60 69 19 1A
:2CD0=ED A0 2B 77 23 78 B1 20
:2CD8=F6 7A 11 00 D8 FE D3 28
:2CE0=E8 D3 E1 C1 D1 E1 F1 C9
:2CE8=CD FB 39 3E 71 32 5D 00
:2CF0=C7 F5 3A 5D 00 07 07 07
:2CF8=07 32 5D 00 F1 C9 E5 21
:2D00=5D 00 CB FE E1 C9 00 00

en 2CC0 ctrl-I (inverse)
en 2CE8 ctrl-N (nettoie)
en 2CF1 ctrl-C (contraste)
en 2CFE ctrl-A (agrandi)
```

```
10 INPUT"CODE ASCii (0-31) :";A
20 PRINTTAB(25)"▣= CTRL ";CHR$($40+A)
30 INPUT"ADRESSE DU PROGRAMME LM ASSOCIE
:";X
40 H=INT(X/256):POKE$67+2*A,X-H*256,H
```

ÉDITEUR DE TEXTES PLEINE PAGE

Ce programme a été mis au point pour permettre à une personne non initiée de taper une lettre sur MZ-80B.

J'ai donc repris quelques peu le système des traitements de textes (ils ont fait leurs preuves !) et j'ai essayé de repenser le principe d'après la facilité d'utilisation de l'éditeur BASIC qui est assez sensationnel.

TAPÉZ VOTRE TEXTE

Ainsi donc vous tapez votre texte et faites la mise en page exactement comme si vous vous trouviez sur une page écran en BASIC. Les fonctions sont reprises de celles utilisées normalement pour l'éditeur :

CLR ; HOME ; CR ; INST ; DEL ; les 4 flèches sont ainsi disponibles pour vous permettre de faire vous même la mise en page. Seules les touches bleues ont une fonction particulière puisqu'elles permettent de sortir de l'éditeur et de l'imprimer.

UTILISATION

Dès son lancement,
le programme demande le nom de la lettre que l'on va écrire.
Si vous ne voulez pas enregistrer le texte, tapez CR ou ENT sans rien écrire d'autre, sinon tapez le nom.
Puis le programme demande le nom du texte à reprendre sur la disquette. Si vous ne voulez pas reprendre de texte, tapez CR ou ENT sans rien écrire d'autre.
Vous avez donc quatre types de réponses possibles :
- soit R le fait de ne rien répondre
- soit N le fait de répondre un nom

NOM DU TEXTE	TEXTE SUR DISQUETTE	EFFET
R	R	vous tapez un texte et vous ne l'enregistrez pas
R	N	vous prenez un texte sur disquette et vous ne l'enregistrez pas
N	R	vous tapez un texte et vous l'enregistrez
N	N	vous prenez un texte sur disquette et vous l'enregistrez

Si vous voulez effacer un texte de la disquette, tapez son nom quand le programme demande le nom de la lettre, et tapez EFFACE et CR ou ENT quand le programme demande le nom de la lettre à reprendre.
En fait, ce programme est plus un essai qu'un programme véritablement utilisable. Il lui manque certaines fonctions qui seraient bien utiles, notamment la création en mode graphique des caractères accentués. Bien sûr il faudra modifier l'entête suivant qui l'utilise, on peut même prévoir un éditeur d'entêtes ! Si vous lui rajoutez de nouvelles fonctions, faites-nous en part.

Marc GIRONDOT

```
5 CONSOLEC80,N:CLR:GOTO66
10 PAGE/P65:PRINT/PCHR$(17);"  ◆◆◆   ◆   ◆  ◆◆◆◆"+CHR$(18)+" AUTOMATISME MESURE
REGULATION"+CHR$(19)
15 PRINT/P" ◆    ◆  ◆◆ ◆◆  ◆   ◆"
20 PRINT/P" ◆    ◆  ◆ ◆ ◆  ◆   ◆              -----------------------------"
30 PRINT/P" ◆◆◆◆◆  ◆    ◆  ◆◆◆◆"
40 PRINT/P" ◆    ◆  ◆    ◆  ◆  ◆"
50 PRINT/P" ◆    ◆  ◆    ◆  ◆   ◆      Rue de la Croix-Vigneron - 95160 MONTMORENC
Y"
55 PRINT/P:PRINT/PTAB(41);"TEL: 989.77.87":PRINT/P
60 PRINT/PTAB(33);"S.A.R.L. AU CAPITAL DE 100.000 F"
61 PRINT/P:PRINT/P:PRINT/P"R.C. SEINE  61 B 2354"
62 PRINT/P:PRINT/P"SIREN  612.023.549"
65 PRINT/PCHR$(16):PRINT/P:RETURN
66 CONSOLEC80:PRINT"NOM DU TEXTE :.................":A=13:C=A:Y$="":D=16:B=0:GOS
UB10000:N$=Y$:IFZ$=CHR$(2)THEN66
67 CONSOLEC80:PRINT"TEXTE SUR DISQUETTE :.................":A=20:C=A:Y$="":D=16
:B=0:GOSUB10000:M$=Y$:IFZ$=CHR$(2)THEN66
68 IFM$="EFFACE"THENON ERROR GOTO66:DELETEN$:GOTO66
69 PRINTCHR$(6);:PRINTTAB(46);"................................."
70 PRINTTAB(46);"................................."
80 PRINTTAB(46);"................................."
90 PRINTTAB(46);"................................."
120 PRINT:PRINT:PRINTTAB(39);"MONTMORENCY, le ..................."
500 CURSOR46,0:A=45:A$="":PRINTSTRING$(".",33);:CURSOR45,0
510 CURSORA+1,0:PRINT".";:GETZ$:IFZ$=""THENCURSORA+1,0:PRINTCHR$(31);:GOTO510
520 IFZ$=CHR$(13)GOTO560
530 IFZ$=CHR$(4)THENIFA<>45THENA$=LEFT$(A$,LEN(A$)-1):CURSORA,0:PRINT"..";:A=A-
1:CURSORA,0
540 IFZ$=CHR$(4)GOTO510
541 IFZ$=CHR$(2)THEN67
545 IFASC(Z$)<32GOTO510
547 IFLEN(A$)=32GOTO510
550 A$=A$+Z$:A=A+1:CURSORA,0:PRINTZ$;:GOTO510
560 IFA$="EFFACE"THENON ERROR GOTO66:DELETEN$:GOTO66
600 D=32:CURSOR46,1:A=45:C=A:B=1:Y$="":PRINTSTRING$(".",33);:CURSOR45,1:GOSUB10
000:B$=Y$:IFZ$=CHR$(2)THENCURSOR46,1:PRINTSTRING$(".",33);:GOTO500
700 CURSOR46,2:A=45:B=2:C=A:Y$="":PRINTSTRING$(".",33);:CURSOR45,2:GOSUB10000:C
$=Y$:IFZ$=CHR$(2)THENCURSOR46,2:PRINTSTRING$(".",33);:GOTO600
800 D=32:CURSOR46,3:A=45:B=3:C=A:Y$="":PRINTSTRING$(".",33);:CURSOR45,3:GOSUB10
000:D$=Y$:IFZ$=CHR$(2)THENCURSOR46,3:PRINTSTRING$(".",33);:GOTO700
900 D=17:CURSOR55,6:A=54:B=6:C=A:Y$="":PRINTSTRING$(".",18);:CURSOR54,6:GOSUB10
000:E$=Y$:IFZ$=CHR$(2)THENCURSOR55,6:PRINTSTRING$(".",18);:GOTO800
1000 CONSOLEC80:ON ERROR GOTO1009:ROPEN#1,FD1@1,M$
1002 FORI=0TO23:INPUT#1,A$:FORJ=1TO80:CURSORJ-1,I:PRINTMID$(A$,J,1);:NEXTJ:NEXT
I:INPUT#1,A$:FORJ=1TO79:CURSORJ-1,24:PRINTMID$(A$,J,1);:NEXTJ
1003 CLOSE
1009 CURSOR0,0
1010 GETF$:IFF$=""THENX=CSRH:Y=CSRV:G$=CHARACTER$(X,Y):CURSORX,Y:PRINTCHR$(31);
:CURSORX,Y:GOSUB1030:GOTO1010
1013 IF(ASC(F$)>15)*(ASC(F$)<26)THEN5000
1014 IFF$=CHR$(1)THEN20000
1015 IFF$=CHR$(2)THEN30000
1016 IFF$=CHR$(3)THEN40000
1017 IFF$=CHR$(4)THEN50000
1018 IF(Y=24)*(X=78)THENPRINTF$;:CURSOR0,0:GOTO1010
1019 IF(Y<>24)*(F$=CHR$(13))THENCURSOR0,Y+1:GOTO1010
1020 PRINTF$;:GOTO1010
1030 IFG$=CHR$(0)THENPRINT" ";:CURSORX,Y:PRINTCHR$(0);:CURSORX,Y:RETURN
1040 PRINTG$;:CURSORX,Y:RETURN
5000 GOSUB10:PRINT/PTAB(45);A$
5010 PRINT/PTAB(45);B$
5020 PRINT/PTAB(45);C$
5030 PRINT/PTAB(45);D$
5050 PRINT/P
5060 PRINT/PTAB(38);"MONTMORENCY, le ";E$
5080 PRINT/P:PRINT/P:PRINT/P
5090 COPY/P1:PRINT/PCHR$(5):IFN$=""THEN5
5091 ON ERROR GOTO5092:DELETEN$
5092 WOPEN#1,FD1@1,N$
6000 FORI=0TO24:A$="":FORJ=0TO79
6005 A$=A$+CHARACTER$(J,I)
6010 NEXTJ:PRINT#1,A$:NEXTI:CLOSE:GOTO5
10000 CURSORA+1,B:PRINT".";:GETZ$:IFZ$=""THENCURSORA+1,B:PRINTCHR$(31);:GOTO100
00
10010 IFZ$=CHR$(13)THENRETURN
10020 IFZ$=CHR$(4)THENIFA<>CTHENY$=LEFT$(Y$,LEN(Y$)-1):CURSORA,B:PRINT"..";:A=A
-1:CURSORA,B
10025 IFZ$=CHR$(2)THENRETURN
10030 IFZ$=CHR$(4)GOTO10000
10035 IFASC(Z$)<32GOTO10000
10037 IFLEN(Y$)=DGOTO10000
10040 Y$=Y$+Z$:A=A+1:CURSORA,B:PRINTZ$;:GOTO10000
20000 INP@234,A:IFA<>239THEN1010
20005 X=CSRH:Y=CSRV:G$=CHARACTER$(X,Y):CURSORX,Y:PRINTCHR$(31);:CURSORX,Y:FORI=
0TO50:NEXT:GOSUB1030
20006 IFY=24THEN20000
20010 PRINTCHR$(1);:GOTO20000
30000 INP@234,A:IFA<>247THEN1010
30005 X=CSRH:Y=CSRV:G$=CHARACTER$(X,Y):CURSORX,Y:PRINTCHR$(31);:CURSORX,Y:FORI=
0TO50:NEXT:GOSUB1030
30010 PRINTCHR$(2);:GOTO30000
40000 INP@234,A:IFA<>191THEN1010
40005 X=CSRH:Y=CSRV:G$=CHARACTER$(X,Y):CURSORX,Y:PRINTCHR$(31);:CURSORX,Y:FORI=
0TO50:NEXT:GOSUB1030
40006 IF(Y=24)*(X=78)THEN40000
40010 PRINTCHR$(3);:GOTO40000
50000 INP@234,A:IFA<>223THEN1010
50005 X=CSRH:Y=CSRV:G$=CHARACTER$(X,Y):CURSORX,Y:PRINTCHR$(31);:CURSORX,Y:FORI=
0TO50:NEXT:GOSUB1030
50010 PRINTCHR$(4);:GOTO50000
```

MEM SYSTEM

Un petite erreur a été commise dans un tableau précédent pour la ligne "Input Buffer Ptr", due à une confusion entre le pointeur du buffer Input et le pointeur du curseur dans ce même buffer. L'erreur est aujourd'hui réparée. De plus, la ligne "WAIT Counter" a été oubliée dans le bulletin n° 18, p. 34 et est donc republiée en son intégralité.

J.F.V.

DEFINITION	PC 1245/51/55	PC 1260/61	PC 1401/02	PC 1350	PC 1450	PC 2500
WAIT Counter	C6E5,C6E6	66E5,66E6	46E5,46E6	70B3,70B4	5F40,5F41	70B3,70B4
PREVIOUS OLD @d	C6FE,C6FF	6700,6701	4700,4701	7498,7499	5F51,5F52	7498,7499
BREAK @ddress	C702,C703	6702,6703	4702,4703	749A,749B	5F5E,5F5F	749A,749B
ERROR @ddress	C706,C707	6704,6705	4704,4705	749C,749D	5F62,5F63	749C,749D
CAL_ Ret. @dd	C6B1,C6B2	6635,6636	46B5,46B6	6F20,6F21	5F20,5F21	6DB0,6DB1
X Buffer	C6B5,C6B6	6639,663A	46B9,46BA	70B5,70B6	5F42,5F43	70B5,70B6
A.P.O. Counter	F8B4,F8BB	6680,6681	607D,607E	70B0 a 70B2	707D,707E	70B0 a 70B2
Funct. OPe. Nbr	C6F2	66F2	46F2	6F30	5F30	6DC0
InP. Buff Cursor	C6EA	66EA	46EA	6F38	5F38	6DC8
InPut Buffer Ptr	C6F1	66F1	46F1	6F2A	5F2A	6DBA
Data Stack Ptr	C6F5	66F5	46F5	6F2D	5F2D	6DBD
Funct. Stack Ptr	C6F6	66F6	46F6	6F2E	5F2E	6DBE
String Buff. Ptr	C6F7	66F7	46F7	6F2F	5F2F	6DBF
Function Stack	F800 a F80F	2880 a 288F	6000 a 600F	74A0 a 74AF	7040 a 704F	74A0 a 74AF
Data Stack	F840 a F87F	2080 a 20BF	6010 a 6067	7680 a 76BF	5DC0 a 5DFF	7680 a 76BF

RELOGEUR DU MONITEUR SHARP

Il a été publié dans le n° 9 un programme de moniteur LM qui était indiqué comme relogeable (Bravo I.S.). Une routine de relogement a été publiée dans le n° 11, malheureusement celle-ci était incomplète:

1) La correction pour d'autres localisations est incomplète, le programme n'affichera pas (MONITOR) et les touches / *, etc., ne sont pas redéfinies.

2) Pourquoi placer obligatoirement ce programme en début de mémoire utilisateur ? On peut aussi le placer dans d'autres zones (exemple de &7C01 à &7FFF sur le PC-1500A) ou en fin de zone BASIC. Je vous propose donc de modifier le programme de chargement pour qu'il soit plus complet et plus souple. Attention, pour placer le programme LM en fin de mémoire BASIC, il faut procéder différemment :

Prendre d'abord @ début = &40C5 par exemple et exécuter le programme BASIC.

Faire un transfert d'octets (Une ligne de BASIC !). Puis faire D = adresse choisie (&55A9 par exemple) directement au clavier et enfin RUN 40.

1986 M. DUMONT
et M. CANILLAC

```
1:REM _***LM_:&257(599)_octets
10:DIM A$(0)*70:H=0:K=&257:WAIT 0:ON ERROR GOTO 90
15:INPUT "@_debut_?";D:A=D
20:FOR L=0TO 17:READ A$(0):FOR I=1TO LEN A$(0)-1STEP 2:B=
  ASC MID$ (A$(0),I,1):B=B-&30-(B>&40)*7:PRINT K
25:C=ASC MID$ (A$(0),I+1,1):C=C-&30-(C>&40)*7:POKE A,B*16
  +C:A=A+1:H=H+C*16+B:K=K-1:NEXT I
30:NEXT L:WAIT :IF H<>84321OR K<>0PRINT "Erreur_d^_introd
  uction_!!!":END
40:RESTORE 280:FOR I=0TO 17:READ V:X=256*PEEK (D+V)+
  PEEK (D+V+1):Y=D+X-&40C5:GOSUB 80
50:POKE D+V,X,Y:NEXT I:Y=D+284:GOSUB 80:POKE D+4,Y:POKE D
  +6,X:POKE D+578,Y
60:Y=D+278:GOSUB 80:POKE D+567,Y:POKE D+569,X:PRINT "O.K.
  ":END
80:X=INT (Y/256),Y=Y-256*X:RETURN
90:H=0:GOTO 30
100:DATA "BE41AF6AE168414A0BBEED3BBEE42C9B05B7578B09B7528B
  C8B72099119ABE421BBE41"
110:DATA "B7BE42ABB5DAF9A378F71A5878B57F1E5AD6E978F800BE41
  EC6AE0FDA8BEE42CFD2A8B"
120:DATA "40B71F8B50B7088B76B70A8B53B70B8B74B70C8B45BE42E3
  832AA778F98B25AE78F9FD"
130:DATA "C8F9B5C4A378F71A5878FD8A1EEB78FB015AF75F0185025F
  FFBE42759E63AE78F9884E"
140:DATA "A578F8996A5AC05878EB78F8019E62BE42759EA0EF78F701
  8513BE4275E978F700B508"
150:DATA "FDDA94AE78EC14AE78ED6A7FFDA88802FD2A88089EA1EF78
  F7FF9312E978F700B70B8B"
160:DATA "04EF78F70FBE42756A075688039E30E978FA00BE42249EC9
  E9787E00E9787F0068776A"
170:DATA "4DB5002E8803FD626C75990D9AFDC8F1BE41CFFD8AB90FB7
  0A8102B306B330419A2F2A"
180:DATA "2D2B3D2E3C204D6F6E69746F72203EB500AE78756A01FDA8
  BE4213FD2A8809EF787505"
190:DATA "6A07FDA8BE4213EF787502FD2A880DEF78750355BEED4D55
  BAED4DFDC8B501AE78FA8E"
200:DATA "02FDC8B548AE7875FD8ABEED4DB520BEED4DB526BEED4D5A
  C05878FD98BEE243BE42E3"
210:DATA "930851BEED4D5EC49910BEE42CBE42E3B70D990AFD0A5AEC
  BEED9551BEED95FDC8B9F8"
220:DATA "51FD8AFBA178EDD9AE78F79A4AC448785AEC58786A07FD98
  5454FDA8BEED9551FD2A88"
230:DATA "0AFD0A4518451AA578FA8B12A578FB8B0DFD986A07F58803
  FD1AE978FB009A5AEC5878"
240:DATA "FD985508550A6A07F58803FD1AFD98B5006A09F913548805
  1E4AC04878FD1A6A01FD98"
250:DATA "FDA86A0A55BE41C78806FD2AFD1A88129AB71F8909FD0AFD
  0AFD0ABA40C5B70E8907FD"
260:DATA "E9F00BFDFB9A4ADB48416840FD60F78B134EE19909B73091
  15B73A9118B741911DB747"
270:DATA "9AA4F99A"
280:DATA 1,31,34,37,58,92,131,156,167,216,230,262,305,320,
  383,399,531,553
```

M ....

ADRESSE ....

....

vous prie de bien vouloir lui faire parvenir

☐ Recueil(s) des n° 1 à 5 des bulletins SHARPENTIER au prix de 100 F*

☐ Bulletin(s) SHARPENTIER n° 6 ☐, 7 ☐, 8 ☐, 9 ☐, 10 ☐ au prix de 20 F chacun*

☐ Bulletin(s) SHARPENTIER n° 11 ☐, 12 ☐, 13 ☐ au prix de 25 F chacun*

☐ Bulletin(s) SHARPENTIER n° 14 ☐, 15 ☐, 16 ☐, 17 épuisé, 18 ☐, 19 ☐ au prix de 30 F chacun*

☐ Manuel(s) Langage Machine PC 1500 ☐, PC 1251 ☐. au prix de 180 F chacun*

☐ Manuel de Référence MZ 800 au prix de 240 F l'un*

* Port forfaitaire : **20 F**

Ci-joint un chèque de .... F à l'ordre du **« CLUB DES SHARPENTIERS »**

Date   Signature

SOMMAIRE